UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

Using supervised machine learning to quantify cleaning behaviour

Raúl Afonso Castelhano de Oliveira

Mestrado em Bioestatística

Dissertação orientada por:
Doutor José Ricardo Paula, Investigador Júnior (orientador)
Doutor Nuno Garcia, Professor Auxiliar (co-orientador)

2024

Funding

This master dissertation was supported by FCT – Fundação para a Ciência e Tecnologia, I.P, within the project *ChangingMoods - Ecological role of unsophisticated cognition within cleaning mutualisms in a changing ocean,* PTDC/BIA-BMA/0080/2021, DOI: 10.54499/PTDC/BIA-BMA/0080/2021

Acknowledgements

I would like to express my gratitude to Professor José Paula for admitting me as his master's student and providing me with the opportunity and resource to develop this thesis at the *Laboratório Marítimo da Guia*.

I would also like to offer my appreciation to Professor Nuno Garcia that from day one, accepted the challenge to supervise this thesis.

I would like to express my deepest gratitude to my professor Helena Nunes, for their invaluable guidance and insightful feedback in shaping my thesis and enhancing its quality.

I am particularly grateful for the colleges I worked with at *Laboratório Marítimo da Guia*, specifically the Behavioural Ecology and Evolution Group.

I want to thank my parents and sister for their support, patience, and encouragement throughout this period.

Finally, I would like to thank my girlfriend, for the patience and hearing me complain during all this time.

Abstract

Cleaner fish are known for their cleaning activities, and the quantification of such inter-specific interactions has been done manually using video recordings. Being a very time-consuming task, it is prone to human classification error due to the long analysis time of the video. Besides that, the classification will range from observer to observer. Within this context, this dissertation aims to develop a semi-automated tracking system to observe and record the movement of the cleaner fish (*Labroides dimidiatus*), and its client fish (*Acanthurus leucosternon*) in a 3-dimensional space laboratory setting, followed by the creation of an algorithm that, with this tracking data, automatically classifies the mutualistic interactions. For the semi-automated tracking system, we used a deep-learning tool specialized in structured subjects, with well-defined body parts, the DeepLabCut (DLC). To the extent of our knowledge, DLC has never been deployed in a multi-fish-species setting. We managed to create a model that reliably tracks both fish with low error. Moreover, we designed an algorithm that successfully detects cleaning interactions with 90% of accuracy. Although the algorithm classified approximately 15% of the non-interactions as interactions, this means that in our case, the algorithm classified a total of 25% of the videos as interactions, meaning that the algorithm was able to reduce the total duration of the videos in 75%, reducing the human labour to approximately 25%.

Keywords: *Labroides dimidiatus*, Animal Behaviour, DeepLabCut, Automation, Interaction Classification

Resumo

Os bodiões-limpadores (*Labroides dimidiatus*) são conhecidos pela sua função de limpeza nos recifes de coral, onde inspecionam e interagem com peixes clientes, onde removem e alimentam-se de ectoparasitas. Estes peixes limpadores chegam a interagir com mais de 2000 clientes por dia, e algumas clientes procuram os limpadores até 145 vezes por dia. O comportamento de limpeza abrange uma variedade de atividades, incluindo mordidas de limpeza, sacudidelas por parte no cliente, estimulação tátil, perseguição por parte do cliente, danças publicitárias do limpador, punição e manipulação. Estas atividades refletem a motivação de limpeza e a qualidade da interação. O *Labroides dimidiatus*, um bodião limpador do Indo-Pacífico, é uma das espécies de peixes limpadores mais estudadas no que toca a interações aquáticas mutualísticas. Este peixe serve como modelo para a investigação sobre o desenvolvimento da cooperação interespecífica, visto ser um peixe essencial para o funcionamento dos sistemas de recife de coral.

As análises destas interações interespecíficas têm sido tradicionalmente realizadas processando manualmente gravações de vídeo. No entanto, este método é demorado e pode não ser confiável devido à variação entre observadores e à fadiga devido às longas horas analisando gravações de vídeos, portanto, alguns jornais exigem uma análise de interobservadores para validar a análise.

À medida que que softwares de rastreamento automático e classificação comportamentais estão se tornando cada vez mais importantes para estudos de comportamento animal (incluindo humanos, ratos e aves), há a necessidade de ferramentas *open-source* que possam rastrear e quantificar as interações de limpeza. Tanto quanto nos é dado a conhecer, não existem tais ferramentas de *open-source* que seja capaz de rastrear os peixes e classificar estas interações de limpeza. As ferramentas presentes na literatura, maioritariamente envolver espaços bidimensionais – geralmente ratos, moscas, peixes – por outro lado, parte da literatura que se foca em espaços tridimensionais, trabalham com animais da mesma espécie.

Devido à inovação do estudo, a implementação do projeto apresenta diversos desafios significativos por si só. Os problemas incluem desde início a recolha de dados e a gravação de vídeos adequados para o nosso objetivo, neste contexto, algumas variáveis precisam ser abordadas. As variações no tamanho dos *Labroides dimidiatus* e de seu cliente de aquário para aquário, a disposição do aquário, as variações de luz entre

aquários, o facto de estarmos a trabalhar com um espaço tridimensional, serem duas espécies diferentes, com formas relativamente distintas, e por fim, a própria forma do peixe. Todos estes aspetos são algo que vêm dificultar todo o nosso trabalho, isto porque se não conseguirmos ter um modelo que seja capaz de rastrear os peixes, identificando os pontos desejados de forma precisa, não se conseguirá prosseguir para a parte da classificação comportamental.

O principal objetivo desta tese é utilizar ferramentas de aprendizagem de automática para criar um sistema semiautomático de rastreamento e identificação para interações de limpeza de *Labroides dimidiatus*. Ao automatizar procedimentos manuais trabalhosos, este sistema procura agilizar o processo de análise, mantendo a fiabilidade dos dados.

Além disso, os dados provenientes do sistema de rastreamento semiautomático serão utilizados para um algoritmo projetados para analisar com foco específico as interações mutualísticas. Ao integrar técnicas de aprendizagem automática no fluxo de trabalho de análise, esta tese procura aumentar a eficácia e rapidez da investigação em ecologia marinha. Inicialmente é necessário identificar ferramentas atuais, parametrizar e adaptar com o objetivo de poder rastrear o movimento das duas espécies de peixes. Para esta tarefa, decidimos então usar a ferramenta DLC (DeepLabCut), a qual é uma ferramenta semiautomática de rastreamento animais. Os dados coletados serão então utilizados num algoritmo para analisar interações interespecíficas. Como resultado, forneceremos uma ferramenta (ou uma definição de configuração para uma ferramenta já existente) que auxiliará os investigadores etólogos que estudam bodiões-limpadores, em particular o *Labroides dimidiatus*.

Neste trabalho foram usadas duas câmaras por aquário, uma camara que grava a parte frontal e outra que grava a parte superior, para que possamos replicar uma gravação tridimensional, potenciando assim a informação obtida pelas gravações dos vídeos. Foram utilizados 6 aquários diferentes com diferentes indivíduos. Visto termos dois ângulos diferentes de câmaras, foram treinadas duas redes neuronais distintas e posteriormente analisadas, obtendo resultados bastante positivos com taxas de erro significativamente baixas. Posteriormente, estes dados foram implementados no algoritmo que tem por base a distância entre os peixes e o tempo que passam perto um do outro. Para calcular a distância, além de diferentes distâncias, foram também testadas diferentes estratégias de medição da distância entre os peixes, que tinha por base o uso de diferentes pontos de referência para calcular a distância. Apesar de não termos conseguido diferenciar os diversos tipos de interações mutualísticas, focamo-nos em diferencias

interação de não interação. Os resultados obtidos à primeira vista não são os melhores, mas, contudo, após uma melhor reflexão, chegamos à conclusão que é um excelente passo para a automatização da classificação de interações. O algoritmo com os melhores parâmetros para o nosso estudo, conseguiu identificar 90% das interações totais dos vídeos usados no estudo. Contudo, classificou aproximadamente de 15% das não interações como interações. Apesar de parecer um erro elevado, o algoritmo classificou somente 25% da duração total dos vídeos, como interações. Com isto em mente, mesmo que seja preciso uma verificação por parte de uma pessoa, esta verificação seria no máximo 25% do tempo utilizado caso tivesse que manualmente analisar os vídeos por completo.

Este trabalho contribui para a comunidade científica com uma base de dados de vídeos *raw* e vídeos manualmente anotados de interações mutualísticas de limpeza. Estes vídeos poderão ser utilizados para testar diversas abordagens de rastreamento ou de classificação comportamental, tanto como conjunto de dados de treino, mas também como conjunto de validação de verdade absoluta.

Como trabalho futuro, seria necessário aumentar a base de dados, visto ter sido possível com uma porção total dos vídeos disponíveis. Posteriormente seria interessante passar de um ambiente controlado como os aquários no laboratório, para um ambiente natural. No entanto, esta mudança apresentaria alguns obstáculos, principalmente no que diz respeito à colheita e análise de dados. O desenvolvimento de metodologias capazes de rastrear e analisar interações em ambientes de recife exigiria técnicas inovadoras que incluam câmaras subaquáticas, algoritmos de rastreamento mais avançados e técnicas de processamento de dados robustas.

A colaboração com especialistas em biologia marinha, visão computacional e aprendizagem automática poderia avançar ainda mais no desenvolvimento de ferramentas automatizadas de rastreamento e classificação adaptadas especificamente para o estudo de interações mutualísticas em ecossistemas de recifes de coral. Ao aproveitar a experiência interdisciplinar e os avanços tecnológicos, a investigação futura pode continuar a expandir os limites do conhecimento em ecologia marinha e análise do comportamento.

Palavras-chave: *Labroides dimidiatus*, Comportamento animal, DeepLabCut, Identificação automática

Table of Contents

List of Figures

List of Tables

List of Equations

Acronyms

BORIS - Behavioural Observation Research Interactive Software

DLC – DeepLabCut

IOR – Inter Observer Reliability

KNN - K-Nearest Neighbors

1.Introduction

1.1. Motivation

In coral reefs, cleaner wrasses are well-known for their cleaning activity, where they inspect client fish and remove ectoparasites while feeding on them (A. S. Grutter, 2002). These cleaner fish can interact with more than 2,000 clients daily, and some clients seek out cleaners for interaction up to 145 times a day (A. Grutter, 1996). Cleaning behaviour includes a wide range of activities such as cleaning bites, client jolts, tactile stimulation, chasing, advertising dances, punishment, and manipulation - all of which provide insight into cleaning motivation and interaction quality (Paula et al., 2019).

One of the most researched cleaner fish species is the Indo-Pacific cleaner wrasse, *Labroides dimidiatus*, which has served as a model system for research on the development of interspecies cooperation (Soares, 2017). This fish is crucial for the functioning of coral reef systems (Demairé et al., 2020), thus it is being used so much in research groups such as the one this work was part of.

Traditionally, these interspecific interactions have been analysed through manual video recording processing (Lindburg, 1969; Pollok et al., 2000; Reiss & Marino, 2001). However, this method is time-consuming and can be unreliable due to variation across human observers and fatigue after hours analysing video recordings, thus some journals require an interobserver analysis to validate the data (Anderson & Perona, 2014; Arac et al., 2019; Dell et al., 2014; Gomez-Marin et al., 2014).

As automatic posture tracking and behavioural classification are becoming increasingly important for accurate animal behaviour studies (e.g. humans, mice, and pigeons) (Alghamdi et al., 2015; Arac et al., 2019; Wittek et al., 2022; Zhang et al., 2020), there is a need for open-source software tools that can measure and classify these types of interactions. To the extent of our knowledge, no software tools can track and classify these cleaning interactions. Most of the literature uses a single animal in two-dimensional spaces - typically mice, flies, or zebrafish (Bohnslav et al., 2021a; Gerós et al., 2020; Guilbeault et al., 2021; Jia et al., 2022a)- while most literature that focuses on three-dimensional spaces uses identical animal species (Arablouei et al., 2023; Han et al., 2024; Long et al., 2020).

Providing a semi-automated tool that speeds up the process and pre-indicates possible interactions in a three-dimensional space allows for less human error and less time spent analysing hours of video footage.

1.2. Interobserver reliability

Three fundamental criteria must be satisfied when evaluating animal data: validity, practicality, and dependability. Whether a measure is expressed by the same observer at different times (intra-observer reliability) or at the same moment by multiple observers (inter-observer reliability, or IOR), the concepts of reliability and assessment reproducibility are closely related. The IOR is a crucial component of trustworthy welfare evaluations, particularly when the assessment is conducted using animal-based indicators, which might be vulnerable to subjectivity and skewed by the assessors' empathy for the animals and prior experiences (Meagher, 2009). Nonetheless, IOR is commonly disregarded in animal behavioural research for a variety of reasons, such as disagreements over the best kind of statistic to employ or challenges in coordinating several observers (Kaufman & Rosenthal, 2009).

1.3. Behavioural Classification

Ecology-driven research fields are particularly interested in the evolutionary foundations of animal behaviour, which can be influenced by outside variables like partner availability, territory restrictions, or nutrient shortages. (Anselme & Güntürkün, 2019; Gill & Wolf, 1975), while molecular and experimental biology use behavioural data together with their techniques to study medical diseases like Parkinson's disease and stress in early life (Kravitz et al., 2010; Mundorf et al., 2020).

On-site manual behavioural observations have consistently improved laboratory and natural habitat investigations seeking to quantify animal behaviour and movement. (Lindburg, 1969; Prior et al., 2008; Reiss & Marino, 2001). Despite all of those innovative accomplishments, manual behavioural observations have the following drawbacks and difficulties: They are labour- and time-intensive, but they also contain a certain amount of subjectivity, which could make it difficult to replicate the tests (Dell et al., 2014). The issues resulting from subjectivity may be mitigated by using camera video recording systems. Unlike a direct observation, a video recording ensures the capture of complete and detailed behavioural patterns during the observation period (Tosi et al., 2006).

Nevertheless, examining video recordings with a pencil, paper, and stopwatch method takes a lot of time as well. Furthermore, because of the observer's unpredictable attention, missed detections are still feasible.

Aside from all of these difficulties in analysing behaviour, it is important to remember that behaviour is a complex, dynamic, and multi-dimensional domain, which makes experimenting with novel techniques a prudent course of action. Currently, the following researchers have benefited from recent developments in computer vision technology as well as an upsurge in interest in artificial intelligence applications: Precise datasets of animal behaviour and movement may be produced with less time and effort, and automatic animal tracking reduces the requirement for human labour and minimizes the likelihood of missed detections (Dell et al., 2014).

1.4. Pose Estimation

In the area of computer vision, pose estimation refers to the particular task of localizing the joint regions or set keypoints of an object (such as a fish, car, or human) in an image. It dates back to the early 1990s research projects that focused on motion tracking, facial pattern recognition, and human detection (Yang & Huang, 1994). Pose estimation has significantly decreased in cost and technical difficulty during the past decade, and because of it, its uses have progressively been incorporated into all aspects of our lives, including gesture-based human–computer interaction (Nguyen et al., 2020), assessment and correction of human movement and posture in healthcare and sport applications (Chen & Yang, 2020), social security — detection of adversary actions (Tsiktsiris et al., 2020), gaming and 3D avatar generation in augmented reality environments (Bleiweiss et al., 2010), among others. The development of deep learning, which has resulted in an exponential rise in posture estimation efforts, opened the door to new areas of this field, such as animal pose estimation. Animal pose estimation plays an essential role in learning and understanding animal behaviour (Joska et al., 2021), understanding the migration of wild animals (Bauer & Klaassen, 2013), and even taking care of dogs (Biggs et al., 2020). Pose estimation for humans and animals can have a similar model structure and foundation. Nonetheless, the diversity of animal species and the scarcity of data must be considered in the construction of animal pose assessment models. The models can be further classified into three primary groups: 3D animal mesh recovery animal, 2D animal pose estimation, and 3D animal pose estimation. (Jiang et al., 2022).

1.5. Problem Description

1.5.1. Problem Statement

Scientific researchers have long employed the species *L. dimidiatus*, both in lab settings and in their native habitat. This species is remarkable as it can learn from other fish and can behave dishonestly or honestly depending on the client fish. It is also important to the structure of coral reefs (Côté & Mills, 2020; Mills & Côté, 2010). This study focuses on the cleaner species *L. dimidiatus* and its client fish *Acanthurus leucosternon*. The following problem question encapsulates our problem statement:

Could mutualistic interactions between *L. dimidiatus* and *A. leucosternon* be automatically classified? To achieve this, we propose an intermediary step: semi-automated tracking in a lab setting. Following that, interaction classification will be performed using the data gathered from this tracking model.

1.5.1.1. Tracking system used

We used machine-learning-based tracking software DeepLabCut (DLC) (A. Mathis et al., 2018; Nath et al., 2019), an open-source, free, and user-friendly program that combines deep neural network transfer learning with an effective approach for 2D and 3D markerless pose estimation that yields great results.

1.5.1.2. Challenges

Due to the novelty of the application domain, the project's implementation presents significant challenges in and of itself. This section breaks it down into smaller problems that can be solved on their own, and the combined (sub)solutions of these problems make up the finished product.

The problems that are portrayed include recording videos and gathering data that are appropriate for our objective. In this context, a few parameters need to be addressed: the variations in the size of the *L. dimidiatus* and its client from aquarium to aquarium, the arrangement of the aquarium, and the fact that all three spatial dimensions of data should be considered.

1.5.1.3. Environment

Due to a variety of environmental circumstances, tracking fish in an aquarium for automatic cleaning behaviour classification might provide major hurdles. First of all,

changes in illumination levels within the aquarium might result in changes in visibility. Therefore, tracking algorithms may find it challenging to reliably identify and track the fish. Other items in the tank, such as decorations, plants, or equipment, can also block the view and cause issues for tracking algorithms. These components not only make the fish less visible, but they also present possible sources of inaccuracy for tracking algorithms. Furthermore, due to the dynamic nature of water, tracking becomes more difficult because of the natural reflections and distortions that can skew visual data and make tracking difficult. Due to the intricacies of the surrounding environment, it is imperative that tracking algorithms be robust and customised to meet the unique challenges of laboratory tank settings so the output data can be used to classify fish behaviours accurately and consistently.

1.5.1.4. Three-dimensional space

We must remember that we are dealing with three-dimensional moving animals. As fish vary in size as well, *L. dimidiatus* may occasionally hide behind the larger fish, making it difficult to determine its exact location. To solve this problem, we used preliminary recordings from 2 separate cameras that were simultaneously filming. One is capturing the front view of the aquarium, and the other is capturing the top view of the aquarium. With this in mind, we can use the top view data, the X axis and Y axis, to become the X axis (depth) and Y axis (width) respectably in the three-dimensional space. We can use the Y axis as the Z axis (height) in the three-dimensional space from the front view. With this in mind, we can build a three-dimensional space from the two camera views.

1.5.1.5. Fish

Interactions with fish are particularly difficult to classify as they have a fusiform body structure, which makes it difficult to establish distinct categorisation markers. Fish interactions present unique challenges, in contrast to other species like mice, who are frequently employed in experimental settings and whose interactions may be explained by the relative placements of their limbs, heads, and tails. Their sleek bodies provide few visual indicators to identify individual encounters, therefore positioning information is not always sufficient. The fluidity of underwater movement adds a further level of intricacy, which allows fish interactions to happen quickly and dynamically in three dimensions. As a result, when it comes to fish, conventional methods of interaction

classification might not be sufficient. Instead, creative solutions that take into consideration the particulars of aquatic settings and fish behaviour are needed.

1.5.1.6. Different species

Tracking and categorising interspecific interactions using two different fish species presents an important challenge. The significant differences in size, shape, pigmentation, and behavioural traits across these species exacerbate the challenges. Studying the interactions between multiple species can be challenging, especially when there is limited research available. Most studies focus on a single species, providing detailed information about their behaviour and ecological roles. However, there is a lack of comparative data that researchers need to consider when trying to understand relationships between different species. Automated categorization techniques are not very effective in understanding the wide range of biological profiles, as algorithms find it difficult to generalize over such a diverse array of profiles. As a result, our understanding of ecosystem functioning and community dynamics is limited by the complexity of interspecies dynamics, which is still not fully understood. To bridge this knowledge gap, we need multidisciplinary groups and creative approaches that consider the diversity of interactions between different species.

1.6. Objectives

This thesis aims to develop a machine learning-driven pose-estimation and behavioural classification system to streamline the analysis of cleaning interactions in marine environments. By reducing the need for manual procedures, this system aims to accelerate the research process while ensuring the accuracy and reliability of the data collected.

The semi-automated tracking system will serve as a foundation for algorithms designed to assess collective dynamics, particularly focusing on mutualistic interactions. The integration of machine learning techniques is expected to significantly improve the efficiency and promptness of marine ecological research.

1.7. Contributions

We hope to contribute materially to the computer science community to improve video and image processing and analysis algorithms, as well as to the biology community with a pipeline of software tools that contains both a method and a successful starter configuration, by making our results (raw and labelled data, tests, and results) available to the community.

In summary, our contributions consist of the following:

Aiming to survey existing tools, identify the most effective ones, train and validate new models to track the movements of two distinct fish species effectively. The resulting data, consisting of different tracking points (X, Y, Z) from both fish, will be crucial for analysing interspecies interactions.

Ultimately, this thesis will deliver a new tool that will aid ethologists in studying cleaner wrasses, specifically *L. dimidiatus*. This is projected to establish a robust framework for future developments in animal behaviour analysis systems, particularly those tailored for cleaner wrasse species.

Furthermore, this work was presented an oral communication at the XX Congress of the Portuguese Ethological Society and as a poster at the XIX Congress of the Portuguese Ethological Society, where it was well welcomed by the academic and scientific communities.

This work is currently in the process of being submitted to a journal for publication.

2.Materials and Methods

2.1. Video Acquisition and Video Analysis

2.1.1. Hardware tools for video acquisition

The videos analyzed in this study were recorded in 2018 and were provided by another research study (Ramírez-Calero et al., 2023). As illustrated in Figure 2.1, the recording setup involves three cameras for every two aquariums, which allows for three-dimensional recordings. For each aquarium, one camera captured the front perspective while another camera recorded the top view. Additionally, the top camera was used to record the top view of two aquariums simultaneously. Videos were recorded in 1920px by 1080px at 50 frames per second.

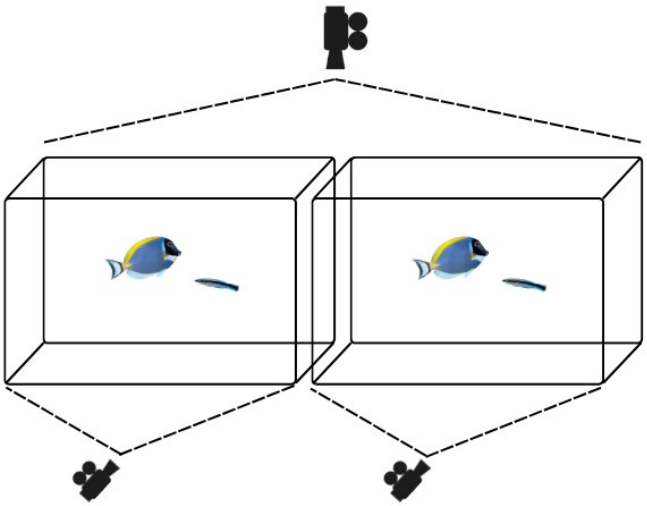

Figure 2.1: Schematic figure depicting the recording setup.

2.1.2. Hardware tools for video analysis

All video analyses were performed on a powerful workstation (ThinkStation P620), with a Processor AMD Ryzen™ Threadripper™ PRO 3945WX @ 4,0 GHz - 4,3 GHz, 32GB of RAM, NVIDIA RTX A4000 16GB RAM, running on a Microsoft Windows 11 Pro.

2.2. Labelling events

In order to use a supervised machine learning approach for the training and testing stages, the interaction events need to be labelled. To accomplish this, we have chosen to use BORIS (Friard & Gamba, 2016), which is an open-source and free software that is easy

to use (Fig. 2.2). we have created an ethogram in Table 2.1, which contains the necessary labels for the interaction events.

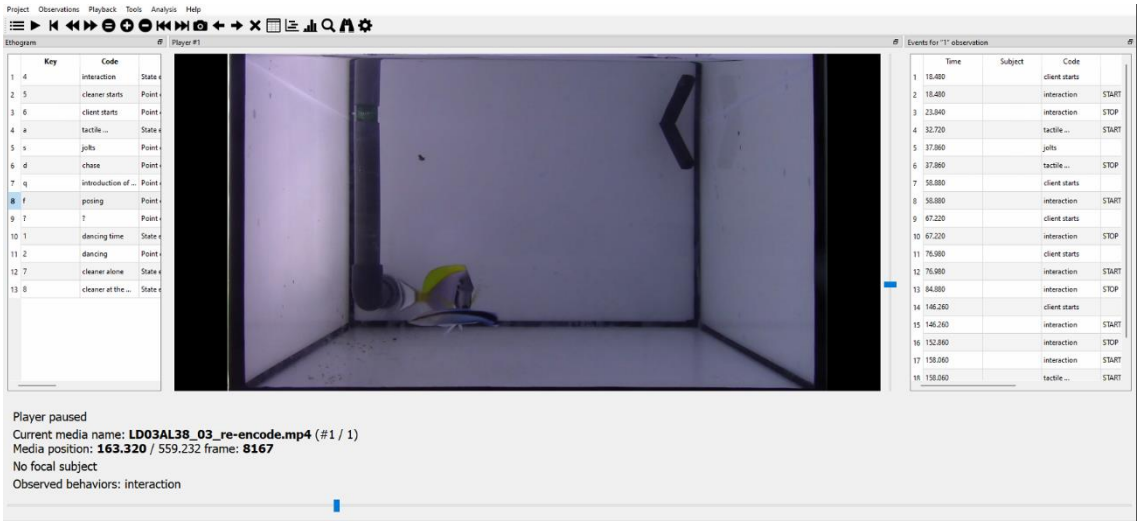

Figure 2.2: BORIS software

Table 2.1: Ethogram of considered behaviour for quantification with BORIS software

Event	Type	Description
Interaction	State Event	Any interaction between the fish
Int_Cleaner	Point Event	Cleaning interaction started by the cleaner
Int_Client	Point Event	Cleaning interaction started by the client
Dance	State Event	Cleaner advertising from the cleaner by dancing
Jolt	Point Event	Client shakes
Posing	Point Event	Client poses
TS	State Event	Cleaner performs tactile stimulation to the client
Chase	Point Event	Client chases the cleaner after an interaction

There are two types of events: State events and Point events. Point events are measured in terms of occurrences, while state events are measured in terms of time (seconds). The interaction events such as Int_Cleaner, Int_Client, Jolt, TS, and Chase involve both fish. Fig 2.3 shows that the dataset became unbalanced due to some interactions that occurred less frequently. We expected some imbalance between interactions and non-interactions, but it was not to this extent. Therefore, we focused on distinguishing between interactions and non-interactions primarily due to this imbalance.

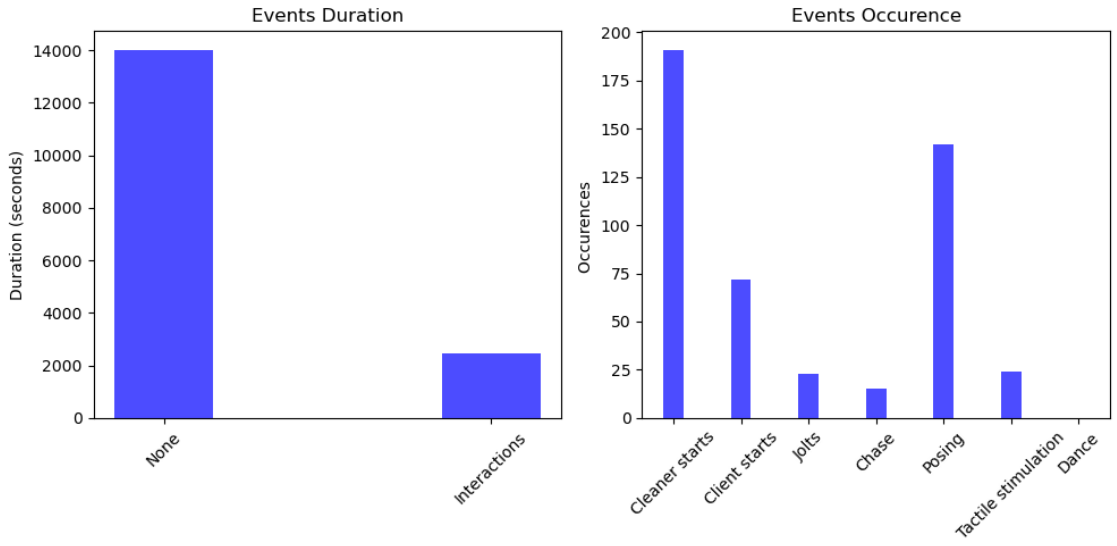

Figure 2.3: Labelled interactions. On the left side we have the comparison between the total time (seconds) of no interactions and all interactions grouped. On the right side we have the comparison between the occurrences of each interaction.

2.3. Pre-Processing Video Data

To make the videos usable in this context, certain modifications were necessary. The original codec of the FrontView movies had to be re-encoded because the tracking tool did not support it. To re-encode, FFmpeg (Bellard, 2023) was utilized, which is a leading multimedia framework capable of decoding, encoding, transcoding, muxing, demuxing, streaming, filtering, and playing nearly everything that humans and machines have developed. The following command line was used:

ffmpeg -i "input_video" -c:v libx265 -preset veryfast -crf 18 "output_video"

We used the Capcut video editing program to split the TopView videos into two parts, one for the left aquarium and the other for the right aquarium. This was based on the camera that was filming each of the two tanks. The edited videos were already in the correct codec, so no further adjustments were needed.

2.4. Tracking tool

In DLC, we used the dlcrnet_ms5 architecture, which is an adaption of two existing ImageNet-pretrained models (ResNets and EfficientNets). DLCRNet_ms5 is inspired by HRNet and use multiple layers to improve the placements of key points and establish connections between them (Lauer et al., 2022), we used the based neural network with default parameters as mentioned in the documentation(M. Mathis, 2021). This decision

was made because we could simultaneously track two distinct species with separate key points, giving the fish different groups. In our instance, we utilised the "uniquebodyparts" group for the *Labroides dimidiatus* tracking points and the "multianimalparts" group for the *Acanthurus leucosternon* tracking points, both are parameters inside the DLC that allow us to do the distinction between two species.

The DLC processing yields multivariate time series data, which are a collection of position values for each body component across time. For example, a 10-minute video yields about 18,000 frames (600 s x 30 fps). This makes our use of DLC a lossy, but semantically enhanced and modified, data reduction process: we convert each frame's raw (decompressed) 1920 x 1080 x 8-bit (16.5 MBit) to 14-body parts (assuming we are labelling a total 14 body parts in total) x 2 x 32 bit (896 bit). This translates to nearly 18,500 reduction factors in the input data's bandwidth.

Since we are trying to replicate a 3D model, we have two different camera views, this means that we need to have two different tracking models.

During the training phase, the DLC model learns to recognize body parts by analyzing labelled images of the animal. These annotations provide ground truth data, which the model uses to map key points and refine predictions using score maps and location refinement algorithms. Once trained, the model can predict and animal's pose in videos by identifying the coordinates of the body parts frame by frames, hence predicting an animal's movement and posture accurately.

In practical terms, this means that given a video as input, the model processes each frame individually and outputs a sequence of vectors. The dimensions of each vector are 2n, where n is the number of body components that are being tracked. Each body parts x and y coordinates match the dimensions. Consequently, the model generates a m×2n matrix for a video consisting of m frames, accurately recording the positions of every tracked body component during the video. This allows for a comprehensive temporal examination of animal behavior and movements, as the coordinates of the fish's key points are tracked throughout the entire video. This data allows us to calculate metrics such as velocity, angles, and, in our specific case, the proximity between fish.

2.4.1. FrontView tracking model

2.4.1.1. Frame Labelling

Frame labelling is a technique used to identify significant elements in an image. In this particular case, the method was applied to recognize the body parts of a fish. We labelled

four distinct keypoints for the *Labroides dimidiatus*, which are the *Cleaner_Mouth*, *Cleaner_Spine1*, *Cleaner_Spine2* and *Cleaner_Tail*, as shown in Figure 2.4.

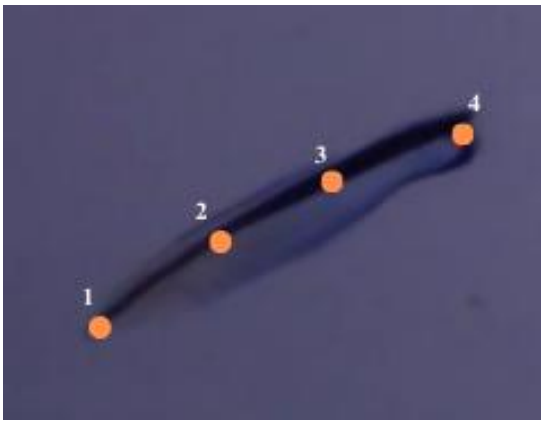

Figure 2.4*: Labroides dimidiatus* labelled FrontView camera keypoints (1- Cleaner_Mouth, 2- Cleaner_Spine1, 3- Cleaner_Spine2 and 4- Cleaner_Tail)

The client fish was labelled with ten distinct key points: Client_Mouth, Client_SpineHead, Client_SpineMid, Client_BodyTop1, Client_BodyTop2, Client_BodyBot1, Client_BodyBot2, Client_Tail, Client_TailTop, and *Client_TailBot*, as shown in Figure 2.5.

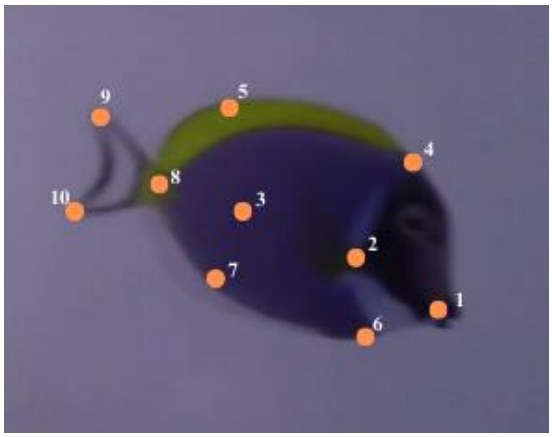

Figure 2.5: *Acanthurus leucosternon* labelled FrontView camera keypoints (1- Client_Mouth, 2- Client_SpineHead, 3- Client_SpineMid, 4- Client_BodyTop1, 5- Client_BodyTop2, 6- Client_BodyBot1, 7- Client_BodyBot2, 8- Client_Tail, 9- Client_TailTop and 10- Client_TailBot

2.4.1.2. Parameters

We labelled 20 frames from each of the twenty videos, for a total of 400 labelled frames for the FrontView tracking model. Out of these, 95% of the frames were used for training. For this, we used a neural network based on dlcrnet_ms5, as recommended by the documentation for multi-animal models. The augmentation method used was imgaug, and the network was trained for 75000 iterations with a batch size of 16.

2.4.2. TopView tracking model

2.4.2.1. Frame Labelling

From the top view cameras, four distinct keypoints were labelled for the *L. dimidiatus* from the top view cameras. These keypoints include the Cleaner_Mouth, Cleaner_Spine1, Cleaner_Spine2, and Cleaner_Tail, as shown in Figure 2.6.

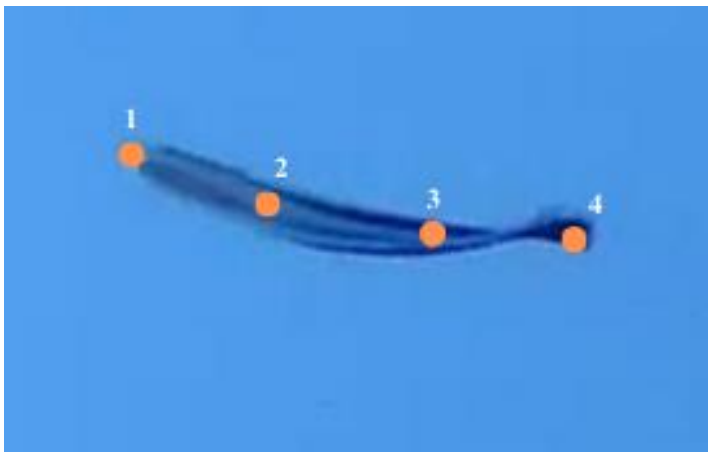

Figure 2.6: *Labroides dimidiatus* labelled TopView camera keypoints (1- Cleaner_Mouth, 2- Cleaner_Spine1, 3- Cleaner_Spine2 and 4- Cleaner_Tail)

Eleven distinct key points were labelled for the client fish, including *Client_Mouth, Client_EyeL*, *Client_EyeR*, *Client_FinBaseL*, *Client_FinTipL*, *Client_FinBaseR*, *Client_FinTipR*, *Client_Spine1*, *Client_Spine2*, *Client_Tail* and *Client_TailTip* as exemplified in Figure 2.7.

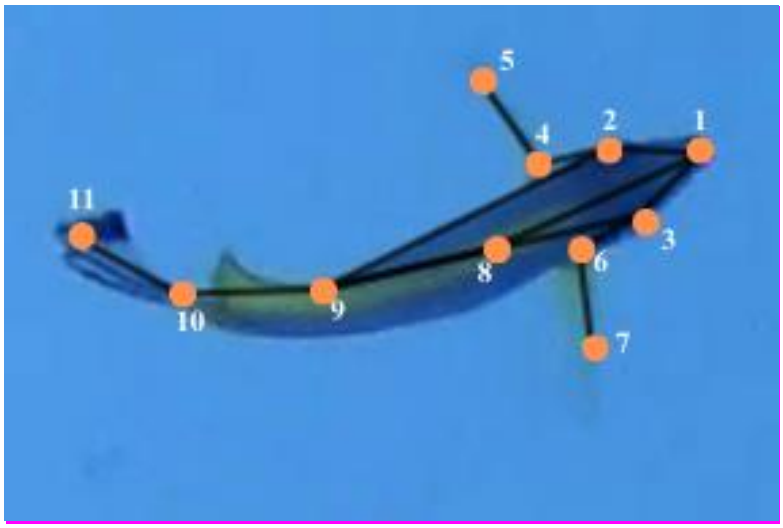

Figure 2.7: *Acanthurus leucosternon* labelled TopView camera keypoints (1- Client_Mouth, 2- Client_EyeL, 3- Client_EyeR, 4- Client_FinBaseL, 5- Client_FinTipL, 6- Client_FinBaseR, 7- Client_FinTipR, 8- Client_Spine1, 9- Client_Spine2, 10- Client_Tail and 11- Client

2.4.2.2. Parameters

We labelled 120 frames from each of the six videos for the TopView tracking model, resulting in a total of 720 labelled frames. We used 95% of these frames for training. To train the model, we utilized a neural network based on the dlcrnet_ms5 architecture. As recommended by the documentation, we used imgaug as the augmentation method. We trained the model for 20000 iterations with a batch size of 16. We reduced the number of iterations compared to the front view model as we noticed that the loss value plateaued similarly to the front view model.

2.4.3. Missing Values

While working with two fish in a 3D environment, it is possible that some body parts may be hidden from view. This could create issues and make my approach unsuitable. To address this problem, we decided to use the K-Nearest Neighbors Imputer (KNNImputer) technique from the scikit-learn machine-learning library (Pedregosa et al., 2011) to impute missing data. We chose this approach because simply deleting or keeping the missing data would not solve the problem. Several factors related to the data characteristics and research goals let to this decision, which will be explained next.

First, the KNN Imputer approach was chosen since it can accurately handle missing values while keeping as much data as possible. Given the intricacy of fish tracking data and the inherent fluctuation in data gathering, keeping the integrity of the study required me to use an approach that maintained as much information as possible.

Second, as the KNN Imputer method can maintain the underlying links between data points, it was judged to be suitable for my tracking data. In tracking fish movements, the temporal and spatial dependencies between observations are essential for understanding behaviour patterns. KNN Imputer preserved the contextual relevance of the imputed values and the relationships within the data by imputing missing values based on the nearest neighbours. Furthermore, the non-parametric structure of KNN Imputer made it appropriate for our tracking data, which may show intricate correlations and distributions.

Moreover, we could adjust the imputation process to the unique features of our tracking data by defining parameters like the weighting method (weights) and the number of nearest neighbours (n_neighbors). The wights parameter determines how much influence each neighbouring observation has on the imputed value. The -_neighbors parameter specifies the number of nearest neighbours to consider during imputation. To

guarantee that the imputation procedure considered enough neighbouring observations and gave equal weight to each neighbour's input, we specified weights='uniform' and n_neighbors=10 in our scenario, this means that we are considering the 10 closest observations, and all neighbours contributed equally.

In summary, the K-Nearest Neighbors Imputer proved to be a suitable option for our analysis of tracking data, providing a harmonious blend of contextual relevance, data retention, and adaptability while managing the intricate nature of fish movement data in an aquarium setting.

2.4.4. Validation

To ensure the accuracy of our results, we can consider multiple factors. Firstly, we can watch the labelled video provided at the end of the model training and check if the dots placed by our DLC model correspond to the fish body parts. This provides a preliminary qualitative analysis of the video and helps us gauge the quality of the output.

Secondly, we can examine the likelihood of the DLC predictions. The likelihood calculation can be mathematically described based on the heatmap output of the neural network.

i. **Heatmap Generation**: For each keypoint, the network outputs a heatmap $H(x,y)$, where (x,y) is the pixel position, and $H(x,y)$ represents the confidence value at that pixel.

ii. **Softmax Normalization**: The raw heatmap values are normalized using the softmax function to ensure they represent probabilities. The normalized heatmap $P(x,y)$ is given by:

$$P(x,y) = \frac{\exp\left(H(x,y)\right)}{\sum_{x',y'} \exp\left(H(x',y')\right)} \quad (1)$$

Where exp is the exponential of H(x,y), and the denomination sums over all pixel positions (x', y') in the exponent of the heatmap, exp(H(x',y')).

iii. **Likelihood Calculation**: The likelihood of a keypoint is the value of the normalized heatmap $P(x,y)$ at the predicted keypoint location $(x_{\max},y_{\max})$, which corresponds to the pixel with the highest intensity in the raw heatmap:

$$Likelihood = P(x_{max}, y_{\max}) = \frac{\exp\left(H(x_{max}, y_{\max})\right)}{\sum_{x',y'} \exp\left(H(x',y')\right)} \quad (2)$$

DLC displays the likelihood of each feature's label for every frame. By calculating the average likelihood of each feature, we can determine if the training was successful - a value greater than 0.90 for every feature indicates success. We can then calculate the average global likelihood of the video by averaging the likelihood of each feature, which is known as the model confidence. When compared to videos with lower values, under 0.95 of likelihood, those with higher values, over 0.95 of confidence, are probably well predicted.

Manual label validation is the third and most reliable method for evaluating model performance. This process involves calculating the Mean Average Error (MAE) between the manually labeled ground truth keypoints (x_i^{true},y_i^{true}) and the model-predicted keypoints (x_i^{pred},y_i^{pred}). The MAE is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i^{true} - x_i^{pred})^2 + (y_i^{true} - y_i^{pred})^2} \quad (3)$$

DeepLabCut integrates this validation approach, allowing for the exclusion of occluded body parts, which often result in high errors and low confidence scores. The computation produces two results: the first considers all labeled keypoints, while the second includes only those with a likelihood value greater than a specified threshold p-cutoff, the default p-cutoff is 0.95, but we decided to lower it to 0.75 to get some lower confident points.

2.5. Behavioural classification

Due to the significant imbalance dataset, as shown before, we decided the best option was to create a function aimed at predicting the start and end of potential interactions. In my approach, we focused on measuring the distance between two fish as the pivotal factor for this prediction, as mutualistic interactions are only plausible when fish are in close proximity to each other. Since our dataset contains many key points, we have analysed various strategies to determine the best method for predicting interactions.

2.5.1. Strategies

2.5.1.1. Strategy 1 - “C4Cleaner-C4Client”

In strategy 1, we calculated the centroid of the cleaner fish by utilizing all 4 keypoints (Cleaner_Mouth, Cleaner_Spine1, Cleaner_Spine2 and Cleaner_Tail), and the centroid of the client fish by using 4 keypoints (Client_Mouth, Client_Spine1, Client_Spine2, Client_Tail). This approach allows the use of other tracking software in the future that utilizes the centroid of the tracking object. Then we calculated the Euclidean distance between the centroids.

2.5.1.2. Strategy 2 - “C4Cleaner-E5Client”

For strategy 2 we calculated the centroid of the cleaner fish using all 4 keypoints (*Cleaner_Mouth, Cleaner_Spine1, Cleaner_Spine2* and *Cleaner_Tail*). Next, we calculated the Euclidean distance to five keypoints from the client fish (*Client_Mouth, Client_Spine1, Client_Spine2, Client_Tail* and *Client_TailTop)* using the cleaner’s centroid and saved the shorter distance. One major reason for modifying Strategy 1 is that since the client is relatively larger, using the centroid as a feature might not be the best option. Therefore, we included the same points as in Strategy 1 along with the client's tail tip, and measured the distance to the closest part.

2.5.1.3. Strategy 3 - “C2Cleaner-E5Client”

In Strategy 3, we followed the same approach as in Strategy 2, but with a slight modification. We reduced the number of key points used in the case of the cleaner. We chose to use only two primary points, namely "Cleaner_Mouth" and "Cleaner_Spine1". These two points hold more significance as the mouth is the primary organ used by the cleaner for its cleaning activity, and the pectoral fins are located between the Cleaner_Mouth and Cleaner_Spine1, and are used by the cleaner to provide tactile stimulation.

We calculated the centroid of the cleaner fish using the two keypoints (*Cleaner_Mouth and Cleaner_Spine1*). Next, we calculated the Euclidean distance to 5 keypoints from the client fish (*Client_Mouth, Client_Spine1, Client_Spine2, Client_Tail* and *Client_TailTop)* using the cleaner’s centroid and saved the shorter distance.

2.5.1.4. Strategy 4 - “C2Cleaner-E10Client”

For Strategy 4, we followed the same approach as Strategy 3, but this time we used all ten keypoints from the client fish. Firstly, we calculated the centroid of the cleaner fish

based on Cleaner_Mouth and Cleaner_Spine1 keypoints. Then, we measured the Euclidean distance from the cleaner's centroid to the 10 keypoints of the client fish, which are Client_Mouth, Client_Spine1, Client_Spine2, Client_SpineTop1, Client_SpineTop2, Client_SpineBot1, Client_SpineBot2, Client_Tail, Client_TailTop, and Client_TailBot. We saved the shortest distance.

2.5.2. Classification equation

The equation works by monitoring a series of distances and identifying when an interaction begins. This happens when the distance drops below a specific threshold. The equation then tracks subsequent frames to ensure that the minimum number of consecutive frames required for an interaction is reached.

If a valid interaction is identified (i.e., when the consecutive frames exceed the threshold), the function logs the start and end frames of that interaction. It also checks whether there is enough margin to combine neighbouring interactions into a single interaction instance.

At the end of the sequence, the equation determines whether an interaction is still in progress and meets the minimum number of consecutive frames requirement. If it does, the function adds it to the observed interactions. The function produces a list of tuples, with each tuple containing the beginning and ending frames of each interaction that was found.

The initial equation takes into account two parameters, minimum consecutive frames, and maximum distance. This makes it reliable and effective for identifying interactions over a series of distances.

$$I_{initial} = \{(i,j) |\ i,j\ \in [1,n], D_i < Max\_Distance, j - i + 1 \geq \text{Min\_}Frames\} \quad (4)$$

Where:

- $I_{initial}$ represents the inicial set of detected interactions
- (i,j) denotes an interaction starting at frame i and ending at frame j
- Di represents the distance at the i -th element of the distance sequence
- The condition Di < *Max_Distance* ensures that the distance at frame i is below the specified threshold

- The condition $(j - i + 1) \geq Min\_Frames$ checks if the number of consecutive frames from i to j is greater than or equal to the minimum consecutive frames required for an interaction

To finalize our equation performed a merging condition. Where:

$$I = TransitiveClosure\ (I_{initial}, \{(i_1, j_1), (i_2, j_2) | i_2 - j_1 \leq margin\}) \quad (5)$$

Where:

- I represent the final set of detected interactions
- **Merging Condition**: For any two pairs $(i_1, j_1), (i_2, j_2) \in I_{initial,}$, if $i_2 - j_1 \leq margin$ and $i_1 < i_2$, merge them into a single pair (i_1, j_2)
- **Update I:** Replace (i_1, j_1) and (i_2, j_2) with (i_1, j_2) in I and repeat until no more pairs satisfy the merging condition.

2.5.2.1. Parameters

A quick exploratory analysis was conducted to determine the distance threshold and interaction length before deciding on the function's parameters. By visualising Fig. 2.8 we gain a better idea of the duration of the interactions in terms of time. We have a good number of interactions with the durations of the interactions less than 2 seconds.

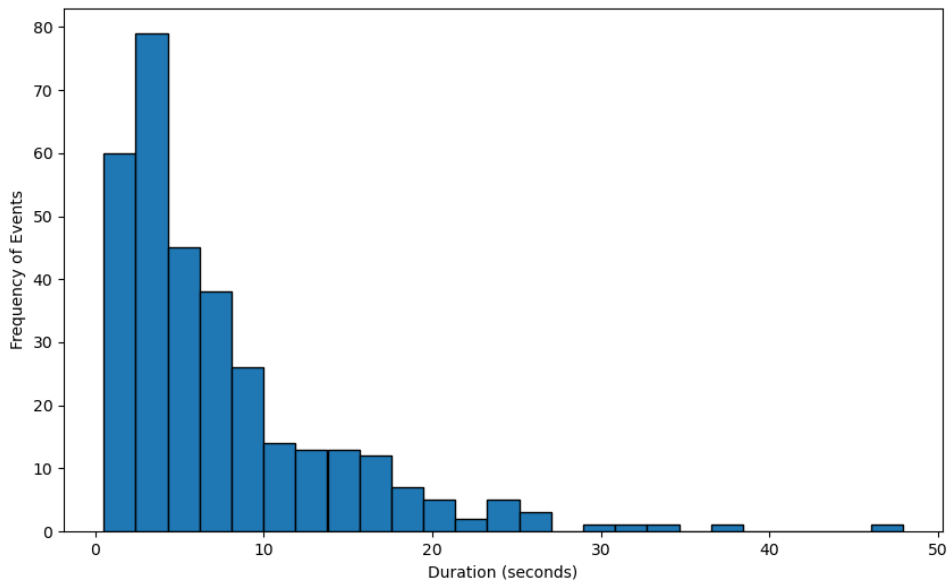

Figure 2.8: Frequency distribution of interaction duration (seconds) of all videos

We need to comprehend more about the frequency of the shorter durations (Table 2.2) and whether there is a possibility that we are losing interactions due to changes in our parameter choices, as my equation depends on the minimal duration while the fish are near one another. For instance, we would lose 43 interactions if we set a minimum time of 60 frames (2 seconds). With this in mind, we choose to set the *Min_Frames* parameter 30, 45, and 60, which corresponds to 1, 1.5 and 2 seconds respectively.

Table 2.2: Interaction Frequency Distribution by Time Intervals

Duration range (seconds)	Occurrences	Cumulative
[0, 1]	5	5
]1, 1.5]	16	21
]1.5, 2]	43	64
]2, ∞ [	284	327

We carried out a comparative distribution of the distances according to each technique for the *Max_Distance* parameter (Figure 2.9). We utilised distinct sets of distances for some strategies since they relied on separate keypoints for measuring distance. For strategy 1 and 2 we used 150, 175, 200, 225 and 250 pixels of distance, while for strategy 3 and 4, we used 100, 125, 150, 175, 200, 225 and 250 pixels of distance. This was mainly due to Strategy 1 and Strategy 2, the points and technique used to measure the distance between the fish being naturally further apart than in the same instance if we used Strategy 3 or 4.

For the *margin* parameter we choose two values: 0 and 30. This means that if there is a gap of less than 30 frames (1 second) between the predicted interaction, and we use the margin = 30, they will be combined into a single interaction; otherwise, if we have margin = 0, this rule does not apply.

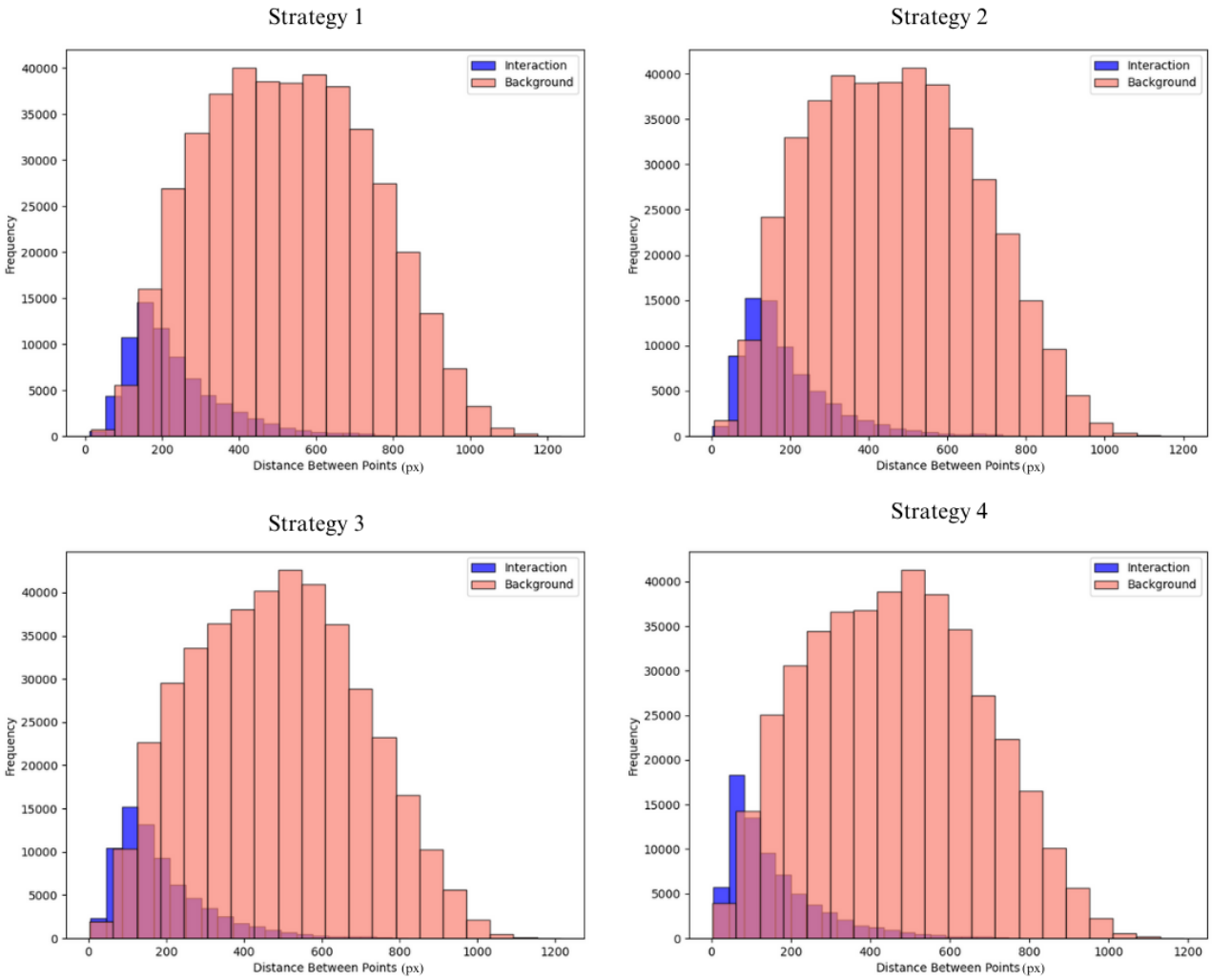

Figure 2.9: Comparative Distribution of Distance Measurements (px) across the four Strategies used, Interaction vs Background

2.5.2.2. Equation Performance

Various indicators were utilized to assess the effectiveness of the classification function in analysing the results of the behavioural classification. Specificity, f1-score, accuracy, precision, and recall were the metrics selected for evaluation. When these metrics are considered in their entirety, they provide valuable information about several aspects of the classification performance and give a robust evaluation of the model's capacity to differentiate between our studied behavioural categories.

To gain a better understanding of the equation, we will examine the total number of correctly and incorrectly predicted interactions, along with the difference between the total number of correctly predicted interactions and the number of duplicate interactions predicted. When the algorithm classifies a single event as multiple events, it results in the prediction of duplicate events. This can occur, for example, if the fish crosses the distance threshold specified in the function. In Figure 2.10, there are a total of three events, but the function classified it as having seven events, leading to the prediction of four duplicate events.

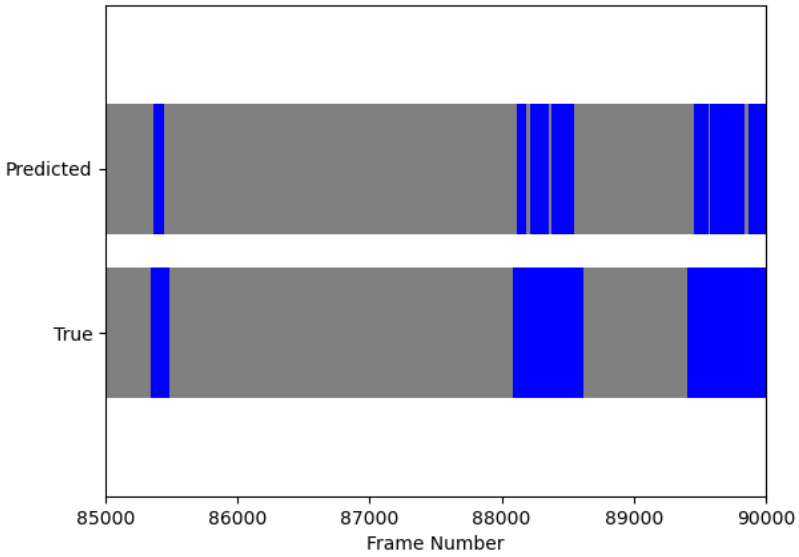

Figure 2.10: Example of duplicate predicted events (blue)

To determine potential differences in performance metrics, a comparative analysis was carried out for each strategy between the parameters Margin = 0 and Margin = 30. To analyze the data, four statistical tests were employed, chosen to address the challenges posed by the small population sizes in this study. These tests were selected following guidelines from prior research on statistical methods appropriate for limited datasets (D'Agostino et al., 1988; Upton, 1982). In these tests, we are comparing the proportions of the Correct Predictions vs Incorrect Predictions according to the data represented in Table 2.3. The statistical tests can be summarised as follows.

Table 2.3: The observed frequencies in a 2x2 Table

	Predicted Correct/ Total correct	***(1-Predicted Correct)/ Total correct***	***Total***
M0	a	c	m
M30	b	d	n
Total	r	s	N

1. **Pearson's Chi-Squared Test**: This test was used to evaluate the independence of categorical variables, providing a baseline measure of association between key variables in the dataset.

$$X^2 = N(ad - bc)^2/rmns \tag{6}$$

2. **Adjusted Pearson's Chi-Squared Test**: Due to the small sample sizes, an adjusted version of the Pearson test was employed to account for potential inaccuracies in the Chi-Squared approximation. This adjustment improved the reliability of the p-values for assessing statistical significance.

$$X_U^2 = (N - 1)(ad - bc)^2/rmns \tag{7}$$

3. **Two Independent Sample t-Test with Pooled Variance**: This test was employed to compare the means of two groups under the assumption of equal variances. The pooled variance approach was chosen to enhance statistical power despite the constraints of small sample sizes.

$$t = \left[\frac{N-2}{N}\right]^{1/2} \frac{ad - bc}{[nac + mbd]^{1/2}} \tag{8}$$

4. **Z-Test for Differences in Proportions**: This test was specifically applied to analyze the proportion of predicted correct outcomes relative to the total corrected outcomes across groups. The z-test was selected for its efficacy in evaluating proportional differences even under limited population conditions.

$$z = \left(\frac{a}{m} - \frac{b}{n}\right) / \sqrt{\frac{s}{N}\left(1 - \frac{S}{N}\right)\left(\frac{1}{m} + \frac{1}{n}\right)} \tag{9}$$

The goal of this test was to identify any notable differences in the efficacy and performance of the model by comparing the proportion of correct predicted under the two Margin settings. This method helped to clarify how changes to the Margin parameter affected the classification results, which helped to guide future configuration tweaks and enhancements to the model.

3. Results

3.1. Filmed Events

It was given 6 videos from the front view and 3 from the top view, which totalled to around 9 hours or 540 minutes of footage and over 250 GB of data. After preprocessing the video data discussed in chapter 2.3, we can now refer to table 3.1, which contains information about the fish ID, duration of the corresponding video, and the camera it was recorded on.

Table 3.1 : Fish IDs and corresponding Metadata

Fish ID	Duration (min)	Camera View
LD03	62	Top
LD04	52	Top
LD13	54	Top
LD14	56	Top
LD23	63	Top
LD24	48	Top
LD03	50	Front
LD04	63	Front
LD13	62	Front
LD14	46	Front
LD23	50	Front
LD24	56	Front

3.2. DeepLabCut results

This section describes the results obtained on the training and validation described in section 2.4. As mentioned in section 2.4, the parameters in Table 3.2 were used for the neural network training, which corresponds to the recommended in DLC documentation, besides batch size, which is related to the power of the workstation in use.

Table 3.2: Parameters used in DLC

Parameters	Values
Training Fraction	0.95
Neural Netwok	Dlcrnet_ms5
Augmentation	Imgaug
P-cutoff	0.95
Track method	Ellipse
Batch size	16

3.2.1. Results of FrontView setup

As described in subsection 2.4.1, the FrontView model was trained for several iterations, specifically 75000, because it was unclear when the training would plateau. However, upon reviewing Fig 3.1, it is evident that 75000 iterations are excessive, as they increase the risk of overfitting and waste valuable time during the model training process.

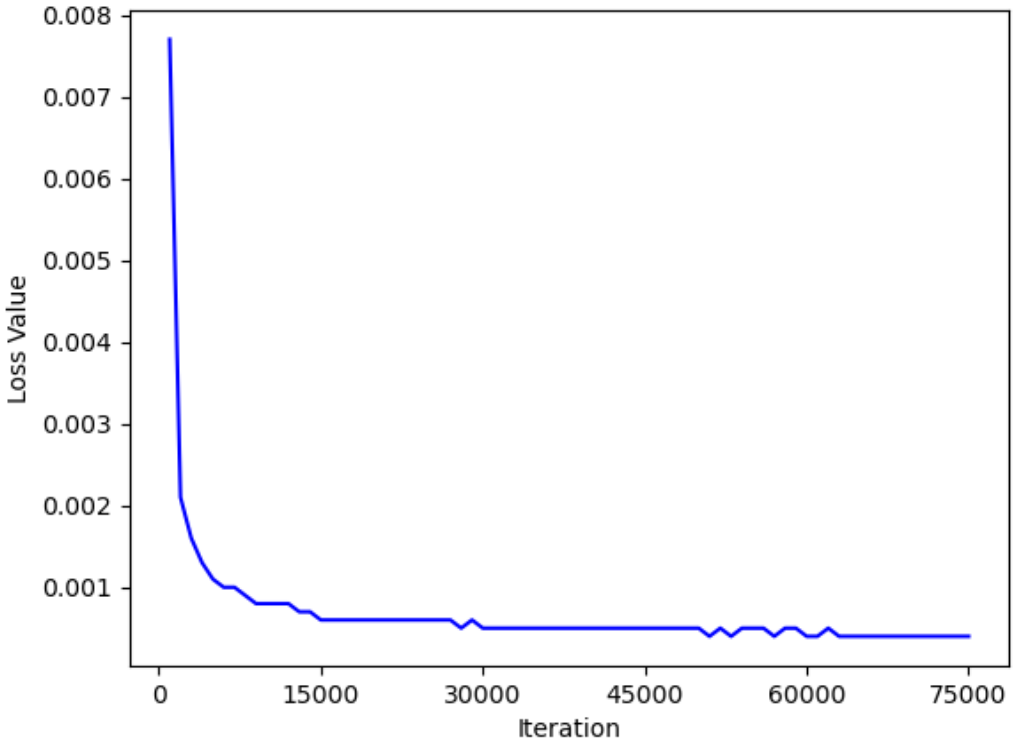

Figure 3.1: FrontView DLC model loss value over the training iterations reflecting the model training progress

After the training was completed, we used the DLC function to "analyze videos and create labelled videos", therefore obtaining the labelled videos with the position of each body part in each frame with its corresponding CSV files. Train and test error summary are displayed in table 3.3. The average likelihood of all the features was 0.92.

Table 3.3: Train and Test error summary at iteration 75000 of the FrontView model. Comparison of Mean Error (px) between Training and Test sets, with and without p-cutoff (0.75)

	Train	Test
Error (px)	4.02	8.66
Error with p-cutoff (px)	4.02	8.32

3.2.2. Results of TopView setup

As explained in subsection 2.4.2, the TopView model was trained for 20000 iterations. Figure 3.2 shows that the loss value decreases rapidly in the first few thousand iterations and then plateaus relatively early.

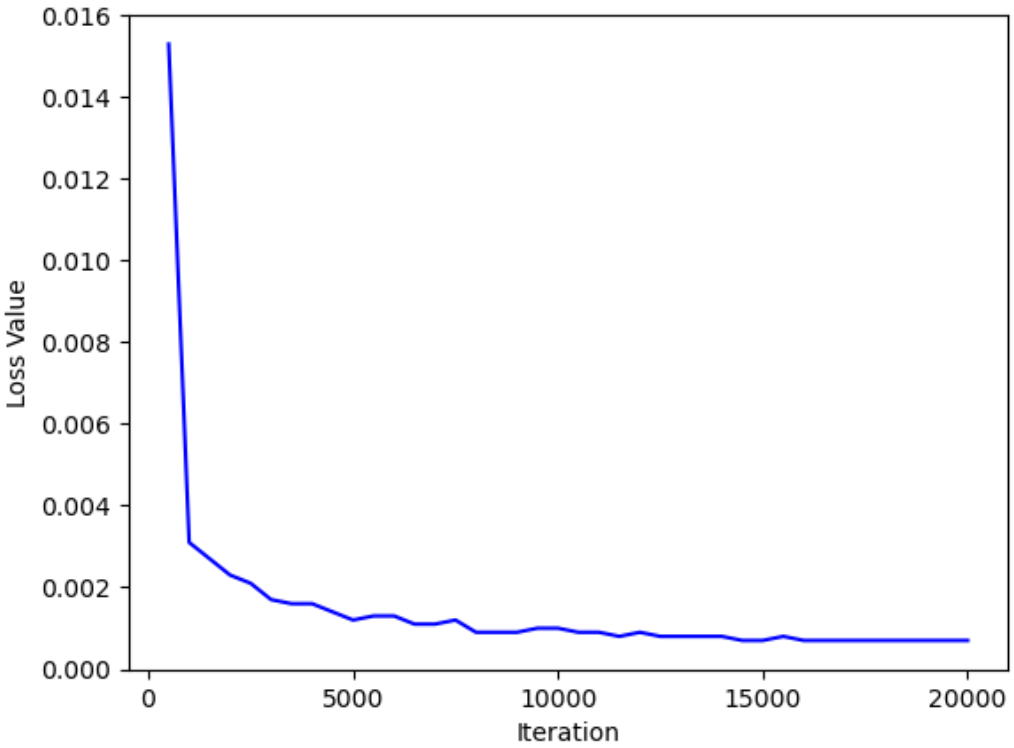

Figure 3.2: TopView DLC model loss value over the training iterations reflecting the model training progress

Figure 3.3: Comparison of strategies (1, 2, 3, and 4) across function parameters (Max_Distance and Min_Frames). Min_Frames (30, 45, and 60) are correspondingly depicted from left to right.Figure 3.4: TopView DLC model loss value over the training iterations reflecting the model training progress

After completing the training, we ran the DLC function “analize videos and create labelled videos”. With these functions, we obtained labelled videos containing the positions of each body part in each frame, along with its corresponding CSV files. We then proceeded with the evaluation process, which resulted in the data presented in table 3.4. The average likelihood for this model was 0.92.

Table 3.4: Train and Test error summary at iteration 20000 of the TopView model. Comparison of Mean Error (px) between Training and Test sets, with and without p-cutoff (0.75)

	Train	Test
Error (px)	4.23	4.73
Error with p-cutoff (px)	4.15	4.56

3.3. Behavioural Classification results

This section describes the results obtained on the different strategies to classify the mutualist interactions as described in sub-section 2.5.2.

3.3.1. Margin Parameter

No statistically significant differences were found between the models with margin = 0 and margin = 30 (p-value > 0.05), suggesting that the choice of margin parameter may not have a substantial impact on model performance in this context (Table 3.5). The only metric that has significant differences is the Correct Interaction Predicted. This finding comes with no surprise, due to the margin parameter was intended to combine events that were separated by less than 1 second. With this in mind, the correct events predicted with margin = 0, will be drastically smaller than those predicted with margin = 30.

Table 3.5: Results of the statistical tests to compare the proportions of correct frames identified with margin = 0 and margin = 30 in terms of the p-values.

Strategy	Pearson's Chi-Squared Test	Adjusted Pearson's Chi-Squared Test	Two Independent Sample t-Test with Pooled Variance	z test for testing the differences in proportions
CC4_CC4	0,929	0,932	0,975	0,972
CC4_C5	0,940	0,942	0,975	0,972
CC2_C5	1,000	1,000	1,000	1,000
CC2_Call	0,940	0,943	0,975	0,971

In light of the findings in Table 3.5, the p-values from each statistical test are consistently extremely high, suggesting that there are no significant differences in the percentages of correct frames found for the strategies examined with margin = 0 or margin = 30. This implies that the margin parameter has no effect on the model's capacity to accurately categorize frames from a statistical standpoint. From a biological perspective, this suggests that the interactions identified by the model are robust to variations in the margin parameter, hence reinforcing the classification strategy's consistency and dependability over a range of thresholds. Because the margin parameter has no apparent impact on the outcomes of interest, we ignore it in the rest of our findings analysis.

3.3.2. Strategies

All metrics mentioned in sub-section 2.5.2.2 are presented in Annex I, which was used for the analysis of function performance. To better visualize and assess the function's

performance with different parameters, multiple plots were created as shown in Fig. 3.3. When solely considering the Min_Frames change, it can be observed that all strategies tend to have lower Precision and Specificity but greater F1-score and Recall with lower Min_Frames.

When focusing on the Max_Distance parameter, it can be observed that Accuracy, F1-Score, and Recall tend to progressively increase when the Max_Distance is maximized. However, the opposite occurs for Specificity and Precision, which tend to gradually drop. The total count of video predictions, as stated in sub-section 2.5.2.2, is shown in Annex II to provide a precise evaluation of the function's performance over the range of parameters. When the Max_Distance parameter is increased, the count of Correct Events, Wrong Events, and subsequently the Correct and Wrong Frames increase overall. On the other hand, when the Min_Frames parameter is increased, the count of Correct Events, Wrong Events, and the predicted Correct and Wrong Frames decreases. Additionally, as Min_Frames increases, predicted duplicate occurrences become less frequent.

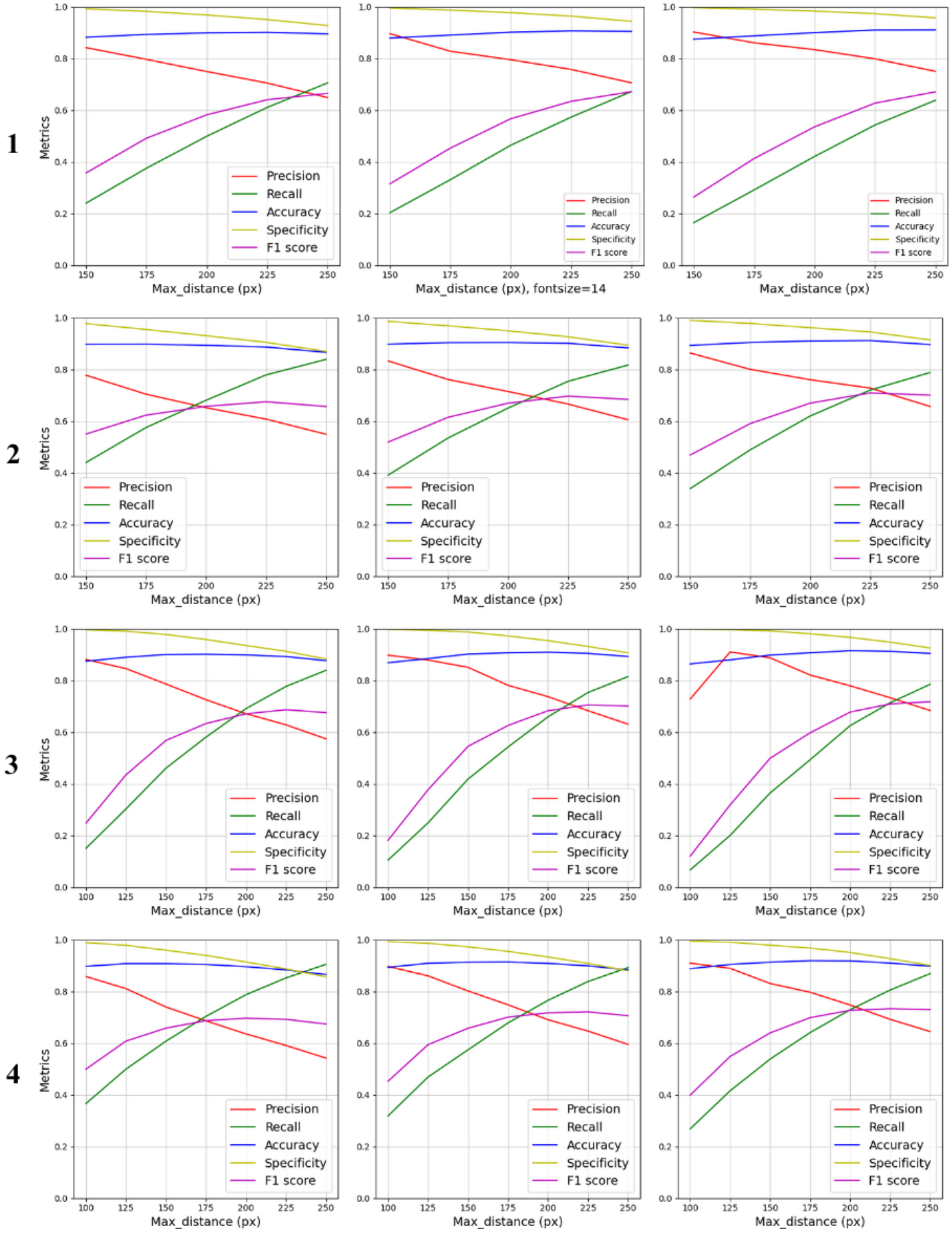

Figure 3.5: Comparison of strategies (1, 2, 3, and 4) across function parameters (Max_Distance and Min_Frames). Min_Frames (30, 45, and 60) are correspondingly depicted from left to right.

4.Discussion

4.1. DeepLabCut

The TopView model consistently outperforms the FrontView model, as evidenced by the test errors detailed in Tables 3.3 and 3.4 and the comparison of average likelihoods across all features. Specifically, Table 3.3 reveals that both the training error and the training error with a p-cutoff were lower than those observed in the test category. This discrepancy was anticipated because the test data were not involved in model training, which was exclusively based on the training data.

Moreover, taking into consideration the FrontView performance (Table 3.3), the reported error of 8.66 pixels is remarkably low, considering it occurs within video footage dimensions of 1920 by 1080 pixels. While the average likelihood of 0.92 may not initially seem ideal, it is important to consider the three-dimensional tracking context. In scenarios where one fish moves in front of another, as depicted in Figure 4.1B, there is a potential for occlusion, where the front fish obscures the one behind. Additionally, natural movements such as turning or swimming away from the camera can hide certain body parts being tracked, such as the mouth shown in Figure 4.1A. These factors can affect the likelihood scores but are typical challenges in three-dimensional tracking environments.

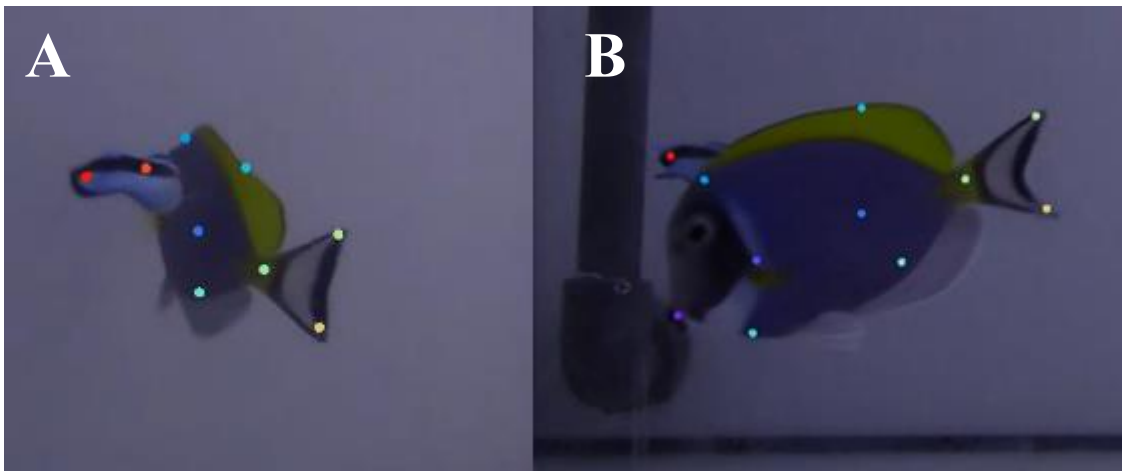

Figure 4.1: FrontView untraceable examples. A- Both fish swimming away from the camera. B- Moments before the *Labroides dimidiatus* being occluded for brief moments.

Considering the performance of the TopView model as shown in Table 3.4, the errors are even smaller than previously noted, which aligns with expectations due to fewer chances for occlusion of the fish's targeted body parts. The average likelihood across all features stands impressively high at 0.96. It is worth noting that even with such high likelihoods, transient occlusions of some body parts still occur multiple times per video, as illustrated

by the example of *Labroides dimidiatus* in Figure 4.2, where the fish spends about 10 minutes obscured beneath the overflow system.
In comparison similar studies (Wiltshire et al., 2023; Wittek et al., 2022) using DLC as a tracking tool, we manage to achieve a similar pixel error in our models, with this in mind we can say that our model proved to be successful.

Further testing of the models on novel videos and subsequent visual analysis indicates robust tracking performance, akin to that observed during model training (https://t.ly/NU8-A). These results from the Deep Learning Components (DLC) are highly satisfactory. Overall, we are pleased with the performance of the system, affirming its effectiveness in real-world applications. performed.

Figure 4.2*: Labroides dimidiatus* hidden under the overflow system, making it difficult to track

4.2. Behavioural Classification

After thorough analysis and review of the results from the behavioural classification, several patterns and conclusions have been identified, specifically related to how various parameters affect the performance of our model.

Conducting a thorough analysis to Figure 3.3, as anticipated, increasing the Max_Distance parameter, which allows a greater distance between fish to be classified as an interaction, reduces both Precision and Specificity. This decrease is due to the model

incorporating more false positives as it misclassifies non-interactive frames as interactive ones when fish are merely within proximity and not truly interacting. However, this flexibility also means that the Recall metric improves because the model is able to capture more true interactions, even those occurring at greater distances between the fish.

Regarding Accuracy, it generally improves with a larger Max_Distance as the model's flexibility in predicting interactions increases. Nonetheless, there may be a slight decline at higher Max_Distance values due to the accrual of false positives which could negate some gains from correctly predicted interactions.

Increasing the Min_Frames parameter, which requires a longer duration of proximity between fish to classify frames as interactions, enhances both Precision and Specificity. This stricter criterion helps minimize false positives, thereby increasing Precision. Specificity also improves because the model is less likely to mistakenly label non-interaction frames as interactions, adhering to the requirement for a longer interaction duration.

Choosing the optimal strategy and its parameters involves balancing these metrics. The F1-score, which harmoniously integrates Precision and Recall, serves as a reliable measure for assessing the overall effectiveness of different strategies. It helps in selecting the strategy that provides the most balanced performance across various metrics.

Considering the trade-offs and performance metrics, a strategy involving a lower Min_Frames and a higher Max_Distance might be the most practical. This configuration allows the algorithm to capture the broadest spectrum of genuine interactions, which should then be supplemented by human verification to confirm the validity of the algorithm's conclusions. Among the various strategies tested, Strategy 4 emerges as the most effective, consistently predicting the most accurate interaction frames.

These insights guide the selection of the most appropriate parameters and strategies to enhance the accuracy and reliability of interaction predictions in the behavioural classification model.

Comparing with similar studies such as (Bohnslav et al., 2021b; Jia et al., 2022b; Wittek et al., 2022), we could not classify in separate behaviours due to what was mentioned in section 2.2. With this in mind it is not possible to compare the studies within themselves, besides that the studies mentioned before did not work with multiple animals nor 3-dimension spaces, they only focused on a 2-dimension space and with one single animal with distinct behaviours.

5.Conclusions

Using DeepLabCut, we successfully trained two models on a notably complex pose estimation challenge involving multiple fish species, specifically *Labroides dimidiatus* and Acanthurus leucosternon, in various behavioural contexts. This work marks the first successful implementation of pose estimation and tracking across multiple fish species in laboratory settings. The models demonstrated robustness across individuals of varying sizes, different aquarium environments, and under conditions of variable camera positioning.

In preliminary tests with novel videos, the models exhibited strong tracking capabilities, performing comparably to the videos used during training, based on visual inspections. However, in terms of behavioural classification, the project did not fully achieve its intended goal of accurately categorizing different behaviours in the videos. Despite this, we developed an algorithm that, when combined with the fish tracking data, significantly reduces the human effort required for behavioural analysis. Specifically, using Strategy 4 with a Max_Distance of 250, the algorithm reduced the analysis workload from about 500,000 frames (approximately 4 hours and 40 minutes of video) to between 104,000 (approximately 57 minutes) and 126,000 frames (approximately 1 hour and 10 minutes). This reduction translates to approximately 75% less video time needed for analysis, with a minimal loss of interaction data—about 10% of total interactions. These losses could potentially be mitigated if interactions were considered to start earlier or end later than initially marked, thereby capturing a fuller extent of the interactions across hundreds of events.

Furthermore, it is important to consider that manually analyzing a raw one-hour video can take from 45 minutes—with no interactions, an unlikely scenario—to potentially over twice the length of the video if interactions are dense. The methodology developed through this project is anticipated to require significantly less time, enhancing efficiency in behavioural research settings.

For future applications of this method, it will be essential to expand the dataset to capture a broader range of interactions. Enlarging the dataset will enable the exploration of key features that uniquely characterize mutualistic interactions. Additionally, this expanded dataset will provide the opportunity to test alternative methodologies and potentially integrate new features such as velocity, direction, or the angle between fish, which could enhance the accuracy and depth of analysis.

With these enhancements, it would also be intriguing to extend this methodology to natural settings. Applying these refined techniques in a natural environment could provide valuable insights into the dynamics of fish interactions in their native habitats, offering a more comprehensive understanding of their behaviours and ecological roles.

References

Alghamdi, M. A., Khan, M. A., & Almotiri, S. H. (2015). Automatic motion tracking of a human in a surveillance video. *2015 IEEE 1st International Smart Cities Conference, ISC2 2015*. https://doi.org/10.1109/ISC2.2015.7366165

Anderson, D. J., & Perona, P. (2014). Toward a Science of Computational Ethology. *Neuron*, *84*(1), 18–31. https://doi.org/10.1016/J.NEURON.2014.09.005

Anselme, P., & Güntürkün, O. (2019). Incentive hope: A default psychological response to multiple forms of uncertainty. *Behavioral and Brain Sciences*, *42*, e58. https://doi.org/10.1017/S0140525X18002194

Arablouei, R., Wang, L., Currie, L., Yates, J., Alvarenga, F. A. P., & Bishop-Hurley, G. J. (2023). Animal behavior classification via deep learning on embedded systems. *Computers and Electronics in Agriculture*, *207*, 107707. https://doi.org/10.1016/J.COMPAG.2023.107707

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. *Frontiers in Systems Neuroscience*, *13*, 446773. https://doi.org/10.3389/FNSYS.2019.00020/BIBTEX

Bauer, S., & Klaassen, M. (2013). Mechanistic models of animal migration behaviour – their diversity, structure and use. *Journal of Animal Ecology*, *82*(3), 498–508. https://doi.org/10.1111/1365-2656.12054

Bellard, F. (n.d.). *FFmpeg*. Retrieved February 1, 2023, from https://ffmpeg.org/

Biggs, B., Boyne, O., Charles, J., Fitzgibbon, A., & Cipolla, R. (2020). Who Left the Dogs Out? 3D Animal Reconstruction with Expectation Maximization in the Loop. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12356 LNCS*, 195–211. https://doi.org/10.1007/978-3-030-58621-8_12

Bleiweiss, A., Eshar, D., Kutliroff, G., Lerner, A., Oshrat, Y., & Yanai, Y. (2010). Enhanced interactive gaming by blending full-body tracking and gesture animation. *ACM SIGGRAPH ASIA 2010 Sketches, SA'10*. https://doi.org/10.1145/1899950.1899984

Bohnslav, J. P., Wimalasena, N. K., Clausing, K. J., Dai, Y. Y., Yarmolinsky, D. A., Cruz, T., Kashlan, A. D., Chiappe, M. E., Orefice, L. L., Woolf, C. J., & Harvey, C. D. (2021a). DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels. *ELife*, *10*. https://doi.org/10.7554/ELIFE.63377

Bohnslav, J. P., Wimalasena, N. K., Clausing, K. J., Dai, Y. Y., Yarmolinsky, D. A., Cruz, T., Kashlan, A. D., Chiappe, M. E., Orefice, L. L., Woolf, C. J., & Harvey, C. D. (2021b). DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels. *ELife*, *10*. https://doi.org/10.7554/ELIFE.63377

Chen, S., & Yang, R. R. (2020). *Pose Trainer: Correcting Exercise Posture using Pose Estimation*. https://arxiv.org/abs/2006.11718v1

Côté, I. M., & Mills, S. C. (2020). Degrees of honesty: cleaning by the redlip cleaner wrasse Labroides rubrolabiatus. *Coral Reefs*, *39*(6), 1693–1701. https://doi.org/10.1007/S00338-020-01996-6/METRICS

D'Agostino, R. B., Chase, W., & Belanger, A. (1988). The Appropriateness of Some Common Procedures for Testing the Equality of Two Independent Binomial Populations. *The American Statistician*, *42*(3), 198. https://doi.org/10.2307/2685002

Dell, A. I., Bender, J. A., Branson, K., Couzin, I. D., de Polavieja, G. G., Noldus, L. P. J. J., Pérez-Escudero, A., Perona, P., Straw, A. D., Wikelski, M., & Brose, U. (2014). Automated image-based tracking and its application in ecology. *Trends in Ecology and Evolution*, *29*(7), 417–428. https://doi.org/10.1016/j.tree.2014.05.004

Demairé, C., Triki, Z., Binning, S. A., Glauser, G., Roche, D. G., & Bshary, R. (2020). Reduced access to cleaner fish negatively impacts the physiological state of two resident reef fishes. *Marine Biology*, *167*(4). https://doi.org/10.1007/s00227-020-3658-2

Friard, O., & Gamba, M. (2016). BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations. *Methods in Ecology and Evolution*, *7*(11), 1325–1330. https://doi.org/10.1111/2041-210X.12584

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. https://doi.org/10.3758/S13428-020-01381-9/FIGURES/6

Gill, F. B., & Wolf, L. L. (1975). Economics of Feeding Territoriality in the Golden-Winged Sunbird. *Ecology*, *56*(2), 333–345. https://doi.org/10.2307/1934964

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. *Nature Neuroscience 2014 17:11*, *17*(11), 1455–1462. https://doi.org/10.1038/nn.3812

Grutter, A. (1996). Parasite removal rates by the cleaner wrasse Labroides dimidiatus. *Marine Ecology Progress Series*, *130*(1–3), 61–70. https://doi.org/10.3354/MEPS130061

Grutter, A. S. (2002). Cleaning symbioses from the parasites' perspective. *Parasitology*, *124*(7), 65–81. https://doi.org/10.1017/S0031182002001488

Guilbeault, N. C., Guerguiev, J., Martin, M., Tate, I., & Thiele, T. R. (2021). BonZeb: open-source, modular software tools for high-resolution zebrafish tracking and analysis. *Scientific Reports 2021 11:1*, *11*(1), 1–21. https://doi.org/10.1038/s41598-021-85896-x

Han, Y., Chen, K., Wang, Y., Liu, W., Wang, Z., Wang, X., Han, C., Liao, J., Huang, K., Cai, S., Huang, Y., Wang, N., Li, J., Song, Y., Li, J., Wang, G. D., Wang, L., Zhang,

Y., & Wei, P. (2024). Multi-animal 3D social pose estimation, identification and behaviour embedding with a few-shot learning framework. *Nature Machine Intelligence 2024 6:1*, *6*(1), 48–61. https://doi.org/10.1038/s42256-023-00776-5

Jia, Y., Li, S., Guo, X., Lei, B., Hu, J., Xu, X. H., & Zhang, W. (2022a). Selfee, Self-supervised Features Extraction of animal behaviors. *ELife*, *11*. https://doi.org/10.7554/ELIFE.76218

Jia, Y., Li, S., Guo, X., Lei, B., Hu, J., Xu, X. H., & Zhang, W. (2022b). Selfee, Self-supervised Features Extraction of animal behaviors. *ELife*, *11*. https://doi.org/10.7554/ELIFE.76218

Jiang, L., Lee, C., Teotia, D., & Ostadabbas, S. (2022). Animal pose estimation: A closer look at the state-of-the-art, existing gaps and opportunities. *Computer Vision and Image Understanding*, *222*, 103483. https://doi.org/10.1016/J.CVIU.2022.103483

Joska, D., Clark, L., Muramatsu, N., Jericevich, R., Nicolls, F., Mathis, A., Mathis, M. W., & Patel, A. (2021). AcinoSet: A 3D Pose Estimation Dataset and Baseline Models for Cheetahs in the Wild. *Proceedings - IEEE International Conference on Robotics and Automation*, *2021-May*, 13901–13908. https://doi.org/10.1109/ICRA48506.2021.9561338

Kaufman, A. B., & Rosenthal, R. (2009). Can you believe my eyes? The importance of interobserver reliability statistics in observations of animal behaviour. *Animal Behaviour*, *78*(6), 1487–1491. https://doi.org/10.1016/J.ANBEHAV.2009.09.014

Kravitz, A. V., Freeze, B. S., Parker, P. R. L., Kay, K., Thwin, M. T., Deisseroth, K., & Kreitzer, A. C. (2010). Regulation of parkinsonian motor behaviours by optogenetic control of basal ganglia circuitry. *Nature 2010 466:7306*, *466*(7306), 622–626. https://doi.org/10.1038/nature09159

Lauer, J., Zhou, M., Ye, S., Menegas, W., Nath, T., Rahman, M. M., Santo, V. Di, Soberanes, D., Feng, G., Murthy, V. N., Lauder, G., Dulac, C., Mathis, M. W., & Mathis, A. (2022). Multi-animal pose estimation and tracking with DeepLabCut. *Nature Methods*, *19*, 496–504. https://doi.org/10.1101/2021.04.30.442096

Lindburg, D. G. (1969). Behavior of infant rhesus monkeys with thalidomide-induced malformations: A pilot study. *Psychonomic Science*, *15*(1), 55–56. https://doi.org/10.3758/BF03336196/METRICS

Long, L., Johnson, Z. V., Li, J., Lancaster, T. J., Aljapur, V., Streelman, J. T., & McGrath, P. T. (2020). Automatic Classification of Cichlid Behaviors Using 3D Convolutional Residual Networks. *IScience*, *23*(10), 101591. https://doi.org/10.1016/J.ISCI.2020.101591

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. *Nature Neuroscience 2018 21:9*, *21*(9), 1281–1289. https://doi.org/10.1038/s41593-018-0209-y

Mathis, M. (2021, May). *What neural network should I use? (Trade offs, speed performance, and considerations) · DeepLabCut/DeepLabCut Wiki · GitHub*. https://github.com/DeepLabCut/DeepLabCut/wiki/What-neural-network-should-I-use%3F-(Trade-offs,-speed-performance,-and-considerations)

Meagher, R. K. (2009). Observer ratings: Validity and value as a tool for animal welfare research. *Applied Animal Behaviour Science*, *119*(1–2), 1–14. https://doi.org/10.1016/J.APPLANIM.2009.02.026

Mills, S. C., & Côté, I. M. (2010). Crime and punishment in a roaming cleanerfish. *Proceedings of the Royal Society B: Biological Sciences*, *277*(1700), 3617–3622. https://doi.org/10.1098/RSPB.2010.0941

Mundorf, A., Matsui, H., Ocklenburg, S., & Freund, N. (2020). Asymmetry of turning behavior in rats is modulated by early life stress. *Behavioural Brain Research*, *393*, 112807. https://doi.org/10.1016/J.BBR.2020.112807

Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., & Mathis, M. W. (2019). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature Protocols 2019 14:7*, *14*(7), 2152–2176. https://doi.org/10.1038/s41596-019-0176-0

Nguyen, N. H., Phan, T. D. T., Lee, G. S., Kim, S. H., & Yang, H. J. (2020). Gesture Recognition Based on 3D Human Pose Estimation and Body Part Segmentation for RGB Data Input. *Applied Sciences 2020, Vol. 10, Page 6188*, *10*(18), 6188. https://doi.org/10.3390/APP10186188

Paula, J. R., Repolho, T., Pegado, M. R., Thörnqvist, P. O., Bispo, R., Winberg, S., Munday, P. L., & Rosa, R. (2019). Neurobiological and behavioural responses of cleaning mutualisms to ocean warming and acidification. *Scientific Reports 2019 9:1*, *9*(1), 1–10. https://doi.org/10.1038/s41598-019-49086-0

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. *Journal of Machine Learning Research*, *12*, 2825–2830. http://scikit-learn.sourceforge.net.

Pollok, B., Prior, H., & Güntürkün, O. (2000). Development of object permanence in food-storing magpies (Pica pica). *Journal of Comparative Psychology*, *114*(2), 148–157. https://doi.org/10.1037/0735-7036.114.2.148

Prior, H., Schwarz, A., & Güntürkün, O. (2008). Mirror-Induced Behavior in the Magpie (Pica pica): Evidence of Self-Recognition. *PLOS Biology*, *6*(8), e202. https://doi.org/10.1371/JOURNAL.PBIO.0060202

Ramírez-Calero, S., Paula, J. R., Otjacques, E., Ravasi, T., Rosa, R., & Schunter, C. (2023). Neuromolecular responses in disrupted mutualistic cleaning interactions

under future environmental conditions. *BMC Biology*, *21*(1), 1–16. https://doi.org/10.1186/S12915-023-01761-5/FIGURES/5

Reiss, D., & Marino, L. (2001). Mirror self-recognition in the bottlenose dolphin: A case of cognitive convergence. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(10), 5937–5942. https://doi.org/10.1073/PNAS.101086398

Soares, M. C. (2017). The Neurobiology of Mutualistic Behavior: The Cleanerfish Swims into the Spotlight. *Frontiers in Behavioral Neuroscience*, *11*. https://doi.org/10.3389/FNBEH.2017.00191

Tosi, M. V., Ferrante, V., Mattiello, S., Canali, E., & Verga, M. (2006). Comparison of video and direct observation methods for measuring oral behaviour in veal calves. *Italian Journal of Animal Science*, *5*(1), 19–27. https://doi.org/10.4081/IJAS.2006.19

Tsiktsiris, D., Dimitriou, N., Lalas, A., Dasygenis, M., Votis, K., & Tzovaras, D. (2020). Real-Time Abnormal Event Detection for Enhanced Security in Autonomous Shuttles Mobility Infrastructures. *Sensors 2020, Vol. 20, Page 4943*, *20*(17), 4943. https://doi.org/10.3390/S20174943

Upton, G. J. G. (1982). A Comparison of Alternative Tests for the 2 × 2 Comparative Trial. *Journal of the Royal Statistical Society. Series A (General)*, *145*(1), 86. https://doi.org/10.2307/2981423

Wiltshire, C., Lewis-Cheetham, J., Komedová, V., Matsuzawa, T., Graham, K. E., & Hobaiter, C. (2023). DeepWild: Application of the pose estimation tool DeepLabCut for behaviour tracking in wild chimpanzees and bonobos. *Journal of Animal Ecology*, *92*(8), 1560–1574. https://doi.org/10.1111/1365-2656.13932

Wittek, N., Wittek, K., Keibel, C., & Güntürkün, O. (2022). Supervised machine learning aided behavior classification in pigeons. *Behavior Research Methods 2022 55:4*, *55*(4), 1624–1640. https://doi.org/10.3758/S13428-022-01881-W

Yang, G., & Huang, T. S. (1994). Human face detection in a complex background. *Pattern Recognition*, *27*(1), 53–63. https://doi.org/10.1016/0031-3203(94)90017-5

Zhang, C., Li, H., & Han, R. (2020). An open-source video tracking system for mouse locomotor activity analysis. *BMC Research Notes*, *13*(1), 1–6. https://doi.org/10.1186/S13104-020-4916-6/FIGURES/3

Annexes

I. Function Evaluation metrics

Strategy	Min_ Frames	Max_ Distance	Precision	Recall	Accuracy	Specificity	F1-score
Strategy 1	30	150	0,842	0,24	0,882	0,992	0,357
		175	0,796	0,375	0,893	0,982	0,491
		200	0,749	0,499	0,899	0,968	0,582
		225	0,705	0,611	0,901	0,95	0,64
		250	0,648	0,706	0,895	0,927	0,665
	45	150	0,896	0,203	0,879	0,995	0,315
		175	0,828	0,331	0,89	0,987	0,453
		200	0,795	0,464	0,901	0,977	0,567
		225	0,757	0,573	0,906	0,963	0,634
		250	0,705	0,672	0,904	0,943	0,671
	60	150	0,902	0,165	0,874	0,996	0,264
		175	0,86	0,291	0,887	0,99	0,413
		200	0,834	0,421	0,9	0,983	0,535
		225	0,798	0,542	0,91	0,973	0,627
		250	0,75	0,638	0,91	0,957	0,671
Strategy 2	30	150	0,777	0,44	0,897	0,977	0,55
		175	0,704	0,575	0,898	0,954	0,623
		200	0,652	0,68	0,893	0,93	0,657
		225	0,608	0,779	0,887	0,905	0,675
		250	0,549	0,839	0,866	0,87	0,657
	45	150	0,833	0,391	0,898	0,986	0,519
		175	0,761	0,535	0,904	0,969	0,615
		200	0,715	0,651	0,904	0,949	0,669
		225	0,666	0,754	0,901	0,926	0,697
		250	0,605	0,817	0,883	0,894	0,684
	60	150	0,863	0,339	0,893	0,99	0,469
		175	0,8	0,489	0,904	0,978	0,59
		200	0,76	0,62	0,91	0,961	0,669
		225	0,728	0,72	0,912	0,945	0,709
		250	0,657	0,788	0,896	0,914	0,701
Strategy 3	30	100	0,881	0,151	0,874	0,996	0,248
		125	0,846	0,303	0,889	0,99	0,435
		150	0,786	0,461	0,9	0,978	0,568

		175	0,725	0,581	0,901	0,958	0,633
		200	0,671	0,691	0,899	0,935	0,671
		225	0,628	0,777	0,892	0,913	0,687
		250	0,574	0,839	0,876	0,883	0,675
	45	**100**	0,898	0,105	0,868	0,998	0,182
		125	0,879	0,249	0,884	0,994	0,377
		150	0,851	0,418	0,902	0,987	0,545
		175	0,781	0,542	0,907	0,972	0,626
		200	0,737	0,659	0,909	0,954	0,683
		225	0,683	0,753	0,904	0,931	0,705
		250	0,631	0,815	0,893	0,906	0,701
	60	**100**	0,727	0,067	0,864	0,998	0,12
		125	0,91	0,201	0,879	0,996	0,318
		150	0,887	0,365	0,898	0,991	0,499
		175	0,821	0,493	0,907	0,98	0,597
		200	0,779	0,625	0,915	0,966	0,678
		225	0,733	0,713	0,912	0,948	0,709
		250	0,684	0,785	0,904	0,925	0,717
Strategy 4	**30**	**100**	0,858	0,366	0,897	0,989	0,499
		125	0,811	0,5	0,907	0,978	0,608
		150	0,74	0,608	0,907	0,959	0,658
		175	0,686	0,704	0,904	0,939	0,687
		200	0,636	0,787	0,896	0,914	0,696
		225	0,591	0,852	0,883	0,887	0,692
		250	0,542	0,905	0,865	0,856	0,674
	45	**100**	0,897	0,318	0,893	0,993	0,453
		125	0,86	0,469	0,909	0,986	0,594
		150	0,802	0,574	0,913	0,973	0,657
		175	0,748	0,678	0,914	0,955	0,701
		200	0,69	0,766	0,908	0,933	0,717
		225	0,646	0,838	0,899	0,908	0,721
		250	0,595	0,892	0,883	0,879	0,706
	60	**100**	0,909	0,268	0,888	0,995	0,399
		125	0,889	0,416	0,904	0,99	0,548
		150	0,831	0,538	0,913	0,979	0,64
		175	0,797	0,641	0,919	0,968	0,698
		200	0,747	0,729	0,918	0,951	0,727
		225	0,692	0,805	0,91	0,927	0,733
		250	0,645	0,869	0,897	0,901	0,73

II. Function Total Predictions

Strategy	**Min_** *frames*	**Max_** *distance*	*Correct Event*	*Duplicate Event*	*Wrong Event*	*True Event*	*Correct Frames*	*Wrong Frames*	*True Frames*	*Videos Frames*
Strategy 1	**30**	**150**	248	104	63	327	16872	3445	73667	494511
		175	332	159	126		26441	7578		
		200	358	158	224		35306	13515		
		225	378	155	340		43662	21044		
		250	391	156	470		50587	30921		
	45	**150**	169	59	28		14021	2169		
		175	239	96	66		22995	5419		
		200	291	112	114		32829	9583		
		225	306	106	182		40965	15344		
		250	329	116	278		48355	23962		
	60	**150**	115	34	17		11216	1580		
		175	182	63	38		20097	3992		
		200	223	74	62		29288	6959		
		225	263	83	101		38735	11234		
		250	280	81	165		45798	18166		
Strategy 2	**30**	**150**	368	174	185	327	30523	9581	73667	494511
		175	397	177	342		40404	19305		
		200	400	170	479		48304	29484		
		225	401	163	623		55812	40577		
		250	388	146	759		60411	55488		
	45	**150**	269	98	79		26952	5850		
		175	314	116	173		37386	13186		
		200	335	121	258		45923	21481		
		225	354	120	368		54079	31418		
		250	345	107	477		58842	45148		
	60	**150**	193	59	46		23057	4176		
		175	244	69	95		33720	9207		
		200	290	91	154		43591	16205		
		225	303	89	212		51392	23377		
		250	308	82	310		56897	36507		
Strategy 3	**30**	**100**	204	74	35	327	11392	1764	73667	494511
		125	315	143	82		21278	26272		
		150	366	169	189		31681	25668		
		175	406	192	324		40443	24554		
		200	413	184	463		48839	22404		
		225	416	181	563		55100	19552		
		250	404	174	705		59614	15891		
	45	**100**	107	31	15		7870	11764		
		125	206	75	39		17308	18634		

		150	280	105	74		28526	18150			
		175	321	122	160		37352	17594			
		200	344	128	243		46275	16391			
		225	367	135	344		53342	14774			
		250	355	129	429		57767	12096			
	60	**100**	55	13	8		5201	8923			
		125	139	46	19		13843	14416			
		150	208	67	38		24795	14632			
		175	253	79	88		33805	14935			
		200	297	102	137		43825	15104			
		225	309	97	204		50363	15568			
		250	317	100	275		55784	15339			
Strategy 4	**30**	**100**	335	157	92	327	25623	4699	73667	494511	
		125	374	160	174		35167	9185			
		150	405	174	301		43194	17358			
		175	415	175	416		50335	25797			
		200	413	173	569		56566	36630			
		225	393	143	676		61357	48012			
		250	386	146	801		65249	61212			
	45	**100**	233	89	42		21930	2941			
		125	303	111	79		32600	5797			
		150	332	123	145		40486	11630			
		175	362	132	234		48430	19208			
		200	366	130	342		54831	28446			
		225	366	120	429		60336	38980			
		250	359	123	534		64277	51397			
	60	**100**	161	54	25		18246	2072			
		125	223	70	43		28468	3973			
		150	282	92	90		37866	8843			
		175	303	96	125		45442	13570			
		200	317	100	196		52278	20968			
		225	322	92	278		58027	31171			
		250	321	95	353		62498	42144			