PRIFYSGOL
BANGOR
UNIVERSITY

School of Computer Science and Engineering
College of Science and Engineering

Semi-Supervised, Species-Invariant Animal Re-Identification From Unrestricted Video

Samuel L. Hennessey

Submitted in partial satisfaction of the requirements for the
Degree of Doctor of Philosophy
in Computer Science

Supervisor Prof. L. I. Kuncheva

19th September 2025

Acknowledgements

> *We cannot solve problems with the same thinking we used to create them*
>
> **— Albert Einstein**

I would first like to express my sincere gratitude to my supervisor, Lucy, for her continuous support and guidance throughout this process. Her expertise and encouragement have been invaluable at every stage of this work.

Secondly, I would like to thank my family for providing the stability and support that enabled me to maintain focus and perseverance throughout the course of this project.

Finally, I also wish to acknowledge the UKRI Centre for Doctoral Training in Artificial Intelligence, Machine Learning and Advanced Computing (CDT AIMLAC) for funding this project under grant EP/S023992/1. This work would not have been achievable without their financial support.

Statement of Originality

The work presented in this thesis/dissertation is entirely from the studies of the individual student, except where otherwise stated. Where derivations are presented and the origin of the work is either wholly or in part from other sources, then full reference is given to the original author. This work has not been presented previously for any degree, nor is it at present under consideration by any other degree awarding body.

Student:

Samuel L. Hennessey

Statement of Availability

I hereby acknowledge the availability of any part of this thesis/dissertation for viewing, photocopying or incorporation into future studies, providing that full reference is given to the origins of any information contained herein. I further give permission for a copy of this work to be deposited with the Bangor University Institutional Digital Repository, the British Library ETHOS system, and/or in any other repository authorised for use by Bangor University and where necessary have gained the required permissions for the use of third party material. I acknowledge that Bangor University may make the title and a summary of this thesis/dissertation freely available.

Student:

Samuel L. Hennessey

Abstract

Animal re-identification is the process of recognising individual animals across different images or video frames, often captured at varying times or locations. Unlike general object detection, which identifies the presence of an animal, re-identification focuses on distinguishing one specific animal from others of the same species. This task is important in ecological monitoring, wildlife conservation, and behavioural studies, where tracking individuals over time provides insights into movement patterns, social interactions, and health status. Due to differences in appearance, pose, lighting, and occlusion, animal re-identification is a challenging problem that often requires specialised datasets and tailored machine learning techniques.

The emergence of machine learning and computer vision has facilitated the automation of animal re-identification, predominantly through supervised learning frameworks and deep learning architectures that depend on manually annotated datasets to achieve consistent and reliable performance. However, the availability of such datasets remains limited, highlighting the necessity for additional benchmark datasets to support the development and evaluation of novel animal re-identification methodologies. In response to this need, a multi-species animal video dataset was constructed, incorporating bounding boxes, identity labels, and multiple feature representations. This dataset served both to evaluate the proposed methods in this work and to address the scarcity of benchmark resources within the field.

As animal re-identification solutions are typically designed for a bespoke subpopulation of a single species, there is a clear need for the development of generalisable methodologies. In response, this work presents the design of a fully autonomous species-invariant animal re-identification pipeline capable of operating in both online and offline scenarios. As a foundational step, an experimental study was conducted to identify the most reliable feature representation for the benchmark dataset in the context of animal re-identification. The results indicated that simple RGB-based features were effective for animal re-identification across species, and were consequently employed in all subsequent experiments.

The development of a novel object detection paradigm is introduced, combining outputs from object detection and multiple object tracking techniques via intersection over union thresholding and connected component extraction to enhance detection accuracy and reliability. To assess and manage the structural complexity of the dataset, both hierarchical and centroid-based constrained clustering approaches were evaluated, with hierarchical clustering proving more successful due to the presence of elongated cluster formations commonly observed in the data.

Building on these findings, the work presented in this thesis contributed to the conception and development of both offline and online semi-supervised clustering methods. Two offline approaches are proposed. The first employs hierarchical clustering of object tracks derived from object detection and multi-object tracking algorithms, with tracks subsequently merged based on classification outputs and a resubstitution confusion matrix. The second approach uses a semi-supervised clustering ensemble, in which a set of constrained clustering algorithms is applied to generate a library of base partitions, which are then integrated using a cumulative adjacency matrix. In addition, an online method was designed to support real-time video analysis by incorporating spatio-temporal constraints into an incremental clustering framework and a likelihood thresholding mechanism to distinguish between new and existing identities. This enables streamlined updates and summarisation of clusters while maintaining low memory requirements and high re-identification accuracy. All proposed semi-supervised clustering methods were evaluated against state-of-the-art baselines and consistently demonstrated superior performance in achieving species-invariant animal re-identification across the benchmark video dataset.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Glossary of Terms

Common Terminology

Abbreviation	**Meaning**
CL	Cannot-Link (Constraints)
ML	Must-Link (Constraints)
MOT	Multiple Object Tracking
BBs	Bounding Boxes
PCA	Principal Component Analysis
CVIs	Cluster Validity Indices

Metrics

Abbreviation	**Meaning**
ARI	Adjusted Rand Index
NMI	Normalised Mutual Information
AP	Average Precision

Notations

Notation	**Definition**
$\mathcal{X}$	The dataset comprising all data points under consideration.
N	The total number of data points contained in $\mathcal{X}$.
c	The total number of classes in the dataset.
x_i	The i^{th} data point in $\mathcal{X}$.
$\mathbf{x}_i$	The feature vector representation of data point x_i.
d	The dimensionality of the feature vector $\mathbf{x}_i$, i.e., the number of features.
$\mathcal{Y}$	The set of labels associated with the data points.

Notation	Definition
y_i	The predicted label corresponding to data point x_i.
$\mathcal{Y}^{GT}$	The set of ground truth labels for all data points.
y_i^{GT}	The ground truth label corresponding to data point x_i.
$\mathcal{Y}^T$	The set of track labels returned by the MOT algorithm.
y_i^T	The track label assigned to data point x_i.
f	A classifier function that maps feature vectors $\mathbf{x}_i$ to predicted labels y_i.
N^C	The total number of constraints in the dataset.
P^C	The proportion of constraints considered or applied.
τ_{ML}	The intersection-over-union threshold that defines a ML constraint.
$\mathcal{ML}$	The set of ML constraints.
$\mathcal{CL}$	The set of CL constraints.
$\mathcal{D}^{ML}$	The set of detections for which ML constraints are defined.
T	The total number of frames in a video sequence.
L	The temporal length of a video, expressed in seconds.
$\mathcal{V}$	A video stream consisting of an ordered set of frames.
F_t	The t^{th} frame in the video stream $\mathcal{V}$.
$\mathcal{D}_t$	The set of detections observed in frame F_t.
$D_{t,i}$	The i^{th} detection in frame F_t.
M	The number of detections contained in a single frame.
WS	The number of consecutive frames in a temporal window.
$\mathcal{B}$	The set of bounding boxes in a frame or sequence.
$\mathcal{B}^{det}$	The set of bounding boxes produced by an object detector.
$\mathcal{B}^{tr}$	The set of bounding boxes produced by an object tracker.
B_i	The i^{th} bounding box in $\mathcal{B}$.
$\mathcal{T}$	The set of frame indices corresponding to bounding boxes.
t_i	The frame index in which bounding box B_i appears in video $\mathcal{V}$.
$\mathcal{P}$	A partition of $\mathcal{X}$ obtained by a clustering algorithm.
C	The set of clusters induced by partition $\mathcal{P}$.
C_i	The i^{th} cluster in C.
K	The total number of clusters in C.
E	The total number of ensemble members in a clustering ensemble.
n_i	The number of data points contained in cluster C_i.

Notation	**Definition**
n_{ij}	The number of data points shared between clusters C_i and C_j.
μ_i	The multivariate mean (centroid) of cluster C_i.
Σ_i	The covariance matrix of cluster C_i.
δ_i	The number of frames elapsed since a new data point was last added to cluster C_i.
C^L	The subset of clusters that have been assigned at least one data point.
C^U	The subset of clusters that have not yet been assigned any data points.
G	A graph defined as the pair (V, E).
V	The set of vertices in graph G.
E	The set of edges in graph G.
W	The set of weights assigned to edges in E.
v	A vertex element of V.
e	An edge element of E.
w	A weight element of W.
$\mathbf{M}_{m,n}$	A matrix of dimensions $m \times n$.
M_{ij}	The $(i, j)^{\text{th}}$ entry of matrix $\mathbf{M}$.
$\mathcal{L}$	The likelihood function associated with a probabilistic model.
$\mathbf{M}^L$	A matrix in which each entry contains a log-likelihood value.
$\mathcal{H}$	The set of assignments $(D_{t,i}, C_j)$ returned by the Hungarian algorithm.
α	A novelty parameter controlling the influence of new data points on a cluster mean μ_i.
β	A log-likelihood threshold used to determine cluster membership for a data point.

Chapter 1

Introduction

1.1 The Problem

The human ability to detect patterns and derive meaning from them has been fundamental to our evolutionary advancement, allowing us to make sense of data that might initially appear ambiguous or unreliable. Among the most notable of these abilities is re-identification, or the recognition of objects we have encountered before. With the emergence of machine learning, these skills have been effectively transferred to machines, enabling high-precision pattern recognition across a wide range of domains—including human detection and re-identification, vehicle tracking, medical diagnostics, speech and character recognition, and industrial automation. However, animal re-identification remains a relatively under-explored field. Although it shares conceptual similarities with person re-identification, it poses distinct challenges—primarily due to the vast diversity of species, each exhibiting unique biometric characteristics. Consequently, existing solutions tailored to human subjects often prove inadequate when applied to animals. Given the increasing urgency surrounding biodiversity conservation and sustainable agricultural practices, the need for reliable and accurate monitoring systems has never been more critical.

Designing systems that leverage video footage introduces an additional layer of complexity, driven by the unpredictable behaviour of subjects in dynamic environments. Key challenges include object occlusion, low video quality, camera movement, and concept drift. Each of these issues presents distinct challenges, requiring carefully designed approaches to ensure reliable and substantively valuable outcomes.

This thesis addresses these challenges by proposing novel approaches to the problem of animal re-identification.

1.2 Aims and Objectives

This work seeks to advance the creation of a fully autonomous software pipeline for species-invariant animal re-identification using video footage. We propose two distinct approaches: an offline pipeline designed for comprehensive analysis of complete datasets, and an online pipeline intended for real-time application. Both pipelines are illustrated in Figure 1.1. Each contribution presented in this work pertains to a specific component within one of the two pipelines. The components to which contributions have been made are denoted in Figure 1.1 using numbered coloured boxes. In several cases, multiple contributions are associated with a single component within a pipeline. Each chapter outlining a contribution highlights the relevant component by displaying the corresponding numbered coloured box.

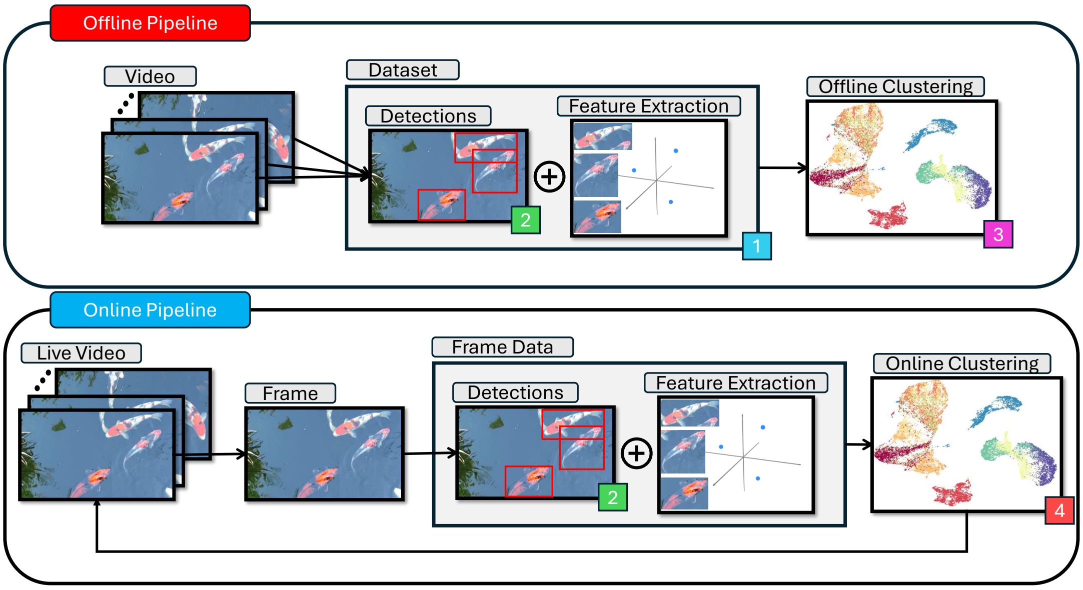


Figure 1.1: Offline and Online versions of our pipeline

To accomplish the aim, the following objectives have been identified

- Create a comprehensive dataset encompassing a diverse range of animal species and incorporating realistic challenges commonly encountered in real-world scenarios, to effectively evaluate the proposed methods.

- Maximise the performance of object detection to reduce the complexity of subsequent animal re-identification.

- Investigate the effectiveness of various clustering techniques applied to datasets with complex spatial structure.

- Develop online constraint-based clustering methods to support real-time animal re-identification.

- Develop offline constraint-based clustering methods tailored to the task of animal re-identification.

1.3 Contributions

This thesis presents work carried out in collaboration with Prof. Ludmila I. Kuncheva, Francis J. Williams, Prof. Juan J. Rodriguez, José Luis Garrido-Labrador and Ismael Ramos-Péres. As a result, the extent of my individual contribution varies across the different projects discussed. The main contributions of this work are as follows:

1. Creating a 5-video dataset including the Ground Truth (GT) annotations (15%). Providing an in depth analysis of the characteristics and challenges involved (85%).

2. Combining bounding boxes output from an object detector and an object tracker as a form of ensemble to improve the overall detection of animals in video frames. (30%)

3. Comparing hierarchical and non-hierarchical clustering for complex data configurations present in animal data. (75%)

4. Proposing an online constrained clustering solution for species-invariant animal re-identification. (85%)

5. Evaluating a constrained clustering ensemble method for clustering a variety of real and synthetic datasets. (35%)

6. Proposing a classification-based clustering method for clustering a range of real-world video datasets. (40%)

1.4 Related Publications

1. L. I. Kuncheva, F. Williams, S. L. Hennessey, and J. J. Rodríguez, “A benchmark database for animal re-identification and tracking,” in 2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS), 2022, pp. 1–6.

2. L. I. Kuncheva, J. L. Garrido-Labrador, I. Ramos-Pérez, S. L. Hennessey, and J. J. Rodríguez, ”An experiment on animal re-identification from video.” Ecological Informatics, 2023, 74, p.101994.

3. L. I. Kuncheva, F. J. Williams, and S. L. Hennessey, “A bibliographic view on constrained clustering,” arXiv preprint arXiv:2209.11125, 2022.

4. F. J. Williams, L. I. Kuncheva, J. J. Rodríguez, and S. L. Hennessey, “Combination of object tracking and object detection for animal recognition,” in 2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS), 2022, pp. 1–6.

5. S. L. Hennessey, F. J. Williams, and L. I. Kuncheva, “Hierarchical Vs Centroid-Based Constraint Clustering for Animal Video Data,” in 2024 IEEE 12th International Conference on Intelligent Systems (IS), 2024, pp. 1–6. *(Winner of the Best Paper Award)*

6. F. J. Williams, S. L. Hennessey, L. I. Kuncheva, J. F. Diez-Pastor, and J. J. Rodríguez, “A Constrained Cluster Ensemble Using Hierarchical Clustering Methods,” in 2024 IEEE 12th International Conference on Intelligent Systems (IS), 2024, pp. 1–6.

7. F. J. Williams, S. L. Hennessey, L. I. Kuncheva, “Animal Re-Identification in Video through Track Clustering” Pattern Analysis and Applications 28, no. 3 (2025): 125.

8. S. L. Hennessey, F. J. Williams, and L. I. Kuncheva, “Real-Time Online Animal Re-Identification from Video using Spatio-temporal Constraints” (*Under Review in Ecological Informatics)*)

1.5 Thesis Overview

Chapter 3 introduces a new benchmark dataset for animal re-identification, alongside an in-depth analysis of the challenges inherent in the data and an experimental study evaluating the generalisability of various feature representations. Chapter 4 outlines a novel object detection paradigm. Chapter 5 presents an experimental investigation into the most effective clustering methodologies for handling the structural complexities of animal video data, and proposes a novel online constrained clustering method for animal re-identification. Chapter 6 introduces two novel offline constrained clustering approaches designed for the same purpose. Finally, Chapter 7 presents a comprehensive discussion of the findings and their implications, along with recommendations for future research.

Chapter 2

Background

2.1 Preliminaries

Animal re-identification constitutes a highly interdisciplinary research area, integrating methodologies and insights from multiple domains that may, at first glance, appear conceptually distinct. As such, certain related fields are addressed only briefly within the scope of this work. To contextualise the multifaceted nature of animal re-identification, we outline the principal areas contributing to its development, as illustrated in Figure 2.1:

- Animal Re-identification
- Multiple Object Tracking
- Classification, Clustering (including constrained and online clustering), Cluster Ensemble and Cluster Validity indices.

2.2 Animal Re-Identification

2.2.1 History of The Problem

The recognition of individual animals predates modern technological developments, with evidence indicating that such practices have been employed for several millennia [28]. Historically, humans relied upon biometric characteristics—including coat patterns, horn morphology, scars, and other distinctive features [66]—as well as artificial markers such as brands and tags, to distinguish between individuals [28]. The advent of photography represented a pivotal advancement, enabling ecologists to systematically document

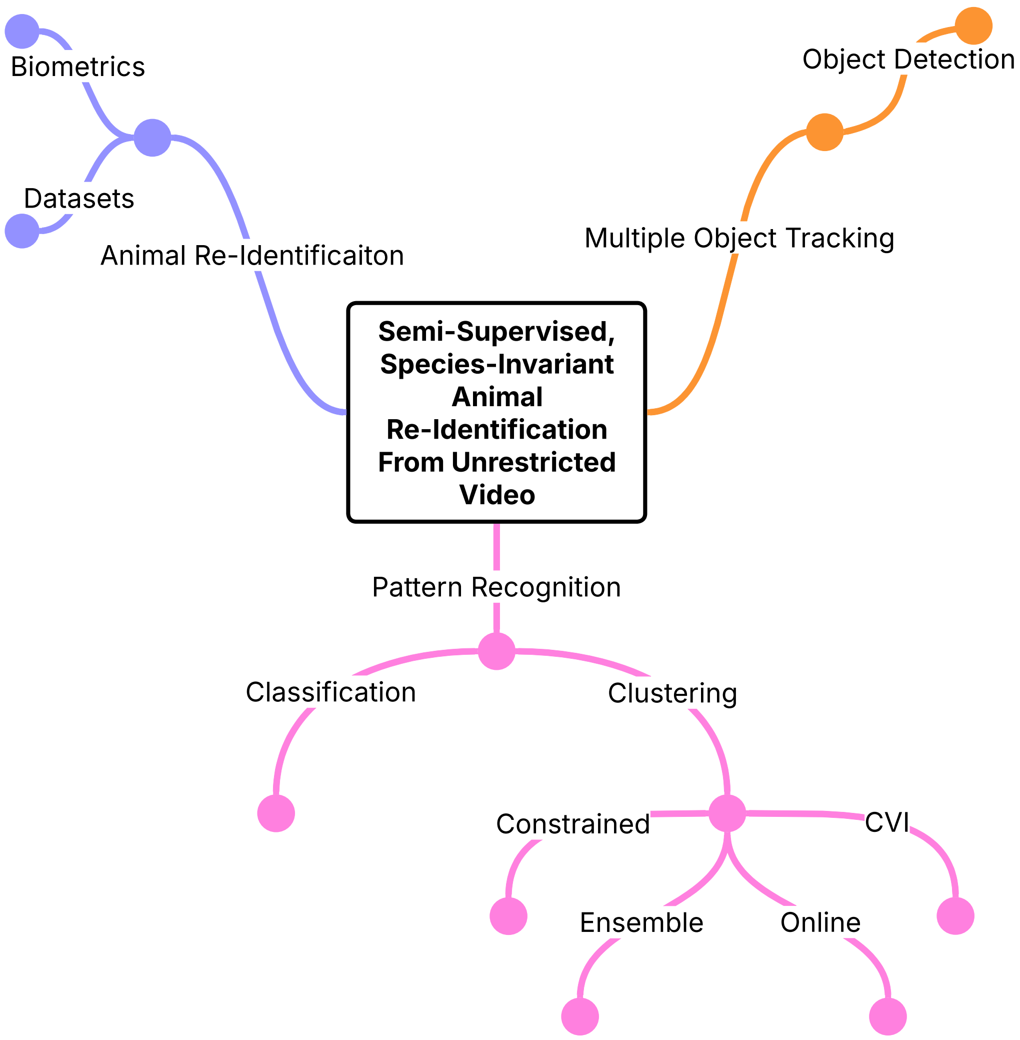


Figure 2.1: Overview of Related Literature Topics

these distinguishing traits. This facilitated more rigorous long-term monitoring and laid the groundwork for contemporary methodologies of animal identification.

By the mid-twentieth century, advances in optical and photographic technologies allowed for more systematic approaches to photo-identification. Researchers identified individual animals manually or semi-automatically, drawing upon distinctive biometric features such as whale flukes, penguin belly patterns, or zebra stripes. A notable milestone occurred in the 1970s when Michael Bigg and colleagues pioneered the photographic identification of killer whales through dorsal fins and saddle patches [26]. In the following decades, computational methods—including feature-matching algorithms such as SIFT [126]—were introduced. The early 2000s witnessed the launch of the Penguin Recognition Project at the University of Bristol, which employed computer vision to non-invasively identify African penguins via their unique chest spot patterns [39], thereby transforming conservation practice by supplanting more intrusive techniques such as flipper bands.

The twenty-first century has seen the proliferation of camera trap technology, generating extensive image datasets. However, conventional feature-engineering methods often proved inadequate under the complex conditions encountered in the field [153]. This limitation has driven the adoption of computer vision and machine learning techniques, many of which were adapted from human re-identification research. Since the mid-2010s, deep learning—particularly convolutional neural networks and metric learning—has catalysed substantial progress in the detection, localisation, and accurate recognition of individual animals [151, 10, 153].

2.2.2 Foundations and Species-Specific Biometrics

Animal re-identification is a field that leverages biometric traits to recognise individual animals that were previously encountered. The ability to re-identify individual animals is not only crucial for population monitoring and conservation but also plays a significant role in advancing behavioural analysis. By reliably tracking individuals over time, researchers, ecologists, and zookeepers can observe patterns of movement, social interaction [154], and health-related behaviours [37, 80, 147, 112, 166]. This enables the development of real-time monitoring systems that support proactive welfare management, allowing for early detection of stress, illness, or changes in routine that may require intervention [165].

The broader discipline of animal biometrics is an emerging area focused on quantifying phenotypic characteristics—such as appearance, behaviour, and morphological features—for species and individuals alike [107, 111]. A foundational example of re-identification through species-specific biometrics involved humans recognising individual swans based on the distinct patterns of their bills [66]. The progression of machine learning, pattern recognition, and computer vision techniques has since enabled this process to be automated with increasing sophistication [169].

Biometric characteristics used in re-identification vary widely across species due to differences in physiology, morphology, and ecological adaptation. In mammals, features such as facial structure [15, 54, 70], coat patterns [206, 38, 208], and even nose prints

[5] are commonly employed. Avian species, in contrast, may rely more heavily on plumage colouration [158] or vocal signatures [107]. Other identifiers like retinal patterns [6, 18], body morphology [78], or gait [1] can also be successful but vary in their applicability and reliability across taxa.

These interspecific differences underscore the need for customised biometric systems tailored to each species' unique traits.

2.2.3 Techniques and Approaches for Animal Re-identification

A critical step in re-identification pipelines is the accurate localisation of relevant image content. Object detection [122, 150] and segmentation techniques allow for the precise isolation of animals from their background and the delineation of their contours [206, 1, 143]. This enables the extraction of visually informative features, such as stripe or spot patterns, facial markings [70, 119], and body shape [205, 36, 95, 39]. By focusing analysis on species-relevant regions of interest (ROIs), these techniques enhance the accuracy and consistency of identification, even under variable environmental conditions [139].

Importantly, the location and type of ROI differ across species [60]. For example, facial features are especially informative in primates [57], while flank patterns or dorsal fins are more relevant for species such as zebras and dolphins [53, 32]. These anatomical and phenotypic differences often limit the performance of generalised re-identification models, necessitating the development of species-specific solutions that can capture and leverage the most discriminative visual cues for each animal.

The features extracted from ROIs—such as texture, colour distribution, shape descriptors, or spatial arrangements—are essential for differentiating individuals. In species with unique markings, such as tigers, giraffes and penguins, localised textures and pattern configurations serve as strong identifiers [39, 36, 53, 158, 160]. In contrast, animals with less distinctive patterns, like elephants or marine mammals, may be more reliably identified by morphological cues such as ear contours [15], dorsal fin contours [78], or skin folds.

Hand-crafted descriptors like Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF) have demonstrated strong performance in animal re-identification, particularly for species with distinctive traits such as fur patterns, skin textures, or body shapes [53, 38, 63, 136, 143]. These methods are resilient to changes in scale, orientation, and lighting, enabling the extraction of consistent local features without the need for extensive training.

On the other hand, deep learning approaches have become central to animal re-identification due to their ability to learn highly discriminative features directly from labelled images [10, 11, 22, 144, 208]. These models are particularly successful at capturing subtle, species-specific traits but typically require large volumes of training data, making them best suited to single-species applications with well-represented datasets. Their performance often declines when applied across taxa, as the learned features may not generalise well without significant retraining [33, 43, 57, 82, 89, 144].

To address data scarcity, recent studies have explored similarity learning techniques [112, 151, 152], such as triplet loss, which embed images into a feature space where visually similar individuals are positioned closer together. This strategy has enabled successful re-identification with limited data across various species, including lions, zebras, chimpanzees, pandas, and tigers [60]. However, its success still depends on the diversity and quality of the training dataset.

Continual learning offers a promising avenue for improving adaptability. Through mechanisms like active learning and human-in-the-loop feedback, models can incrementally update to recognise new individuals and accommodate appearance changes over time [30]. However, it still largely operates within a single-species framework and does not fully address the issue of cross-species generalisation.

2.2.4 Limitations, Alternatives, and Future Directions

Traditional machine learning approaches have demonstrated considerable success in animal re-identification, particularly in scenarios where clear visual patterns—such as stripes, spots, or other biometrics—can be extracted and used for matching individuals. These methods typically rely on feature descriptors and pattern-matching algorithms to compare new images against a database of known individuals [107, 54, 78, 113, 63].

One of their strengths lies in their relative simplicity and ability to generalise across species, especially those with distinct coat or skin patterns. However, these approaches are inherently dependent on the existence of prior examples for each individual in the dataset, limiting their capacity to identify previously unrecorded identities [153].

In situations where animals do not possess naturally distinctive patterns, researchers have occasionally resorted to artificially applying visual markers to facilitate re-identification. This may involve using dyes, tags, or even branding techniques to create unique, identifiable patterns [98, 95, 64, 172]. While successful in controlled environments such as livestock farms or zoos, this approach is not viable for wildlife populations.

The use of a unified system for species-level re-identification has shown promising results, with many models successfully distinguishing between different species of animals from images [49, 67, 186, 135, 139, 150, 170]. These systems typically rely on well-established classification architectures trained on diverse datasets encompassing multiple species. However, this success has not translated to individual-level re-identification across multiple species.

All current state-of-the-art solutions for animal re-identification rely on supervised learning approaches [82, 89], which require large volumes of annotated training data [5, 10, 17, 22, 67]. To date, no dependable unsupervised approach has been developed or widely adopted for animal re-identification. Traditional machine learning techniques—such as keypoint matching or template-based methods—still depend on curated reference data [15, 53, 113, 117, 158, 160].

The development of unsupervised or self-supervised methods could therefore open new avenues for more scalable and adaptable re-identification systems in the future.

2.2.5 Available Datasets

A significant bottleneck in the advancement of animal re-identification systems is the lack of large-scale, high-quality datasets. In contrast to human re-identification, which benefits from a wealth of publicly available and diverse benchmarks, animal datasets are typically limited in scope, species-specific, and often collected in controlled or localised settings [91, 175, 103, 120]. This scarcity stems from the inherent challenges of capturing consistent, labelled images of individual animals in natural environments—where lighting conditions, occlusion, pose variability, and complex backgrounds pose significant obstacles. Furthermore, many animal species lack distinct visual identifiers, making it difficult to compile annotated datasets with sufficient intra-class variation. As a result, models trained on these datasets often face difficulties in generalising to new individuals, settings, or species. Overcoming this limitation calls for collaborative data-sharing efforts, the adoption of data-efficient learning paradigms, and a shift toward unsupervised or few-shot methods capable of functioning successfully under data-scarce conditions.

While efforts have been made to autonomously gather animal imagery from the web for species classification tasks [21], using web crawlers and large-scale data to build general-purpose models across multiple taxa, no analogous method currently exists for individual-level re-identification. The primary reason lies in the lack of labelled web images that associate each animal with a known identity, along with the fine-grained, species-specific biometric features needed for re-identification, which are often absent or inconsistent in web-sourced data. Consequently, individual re-identification remains heavily reliant on curated datasets acquired through manual annotation or structured data collection, some of which are summarised in 2.1. This reliance significantly limits scalability and impedes the development of systems capable of generalising across species or deployment in unconstrained environments.

Table 2.1: A table of available databases for animal re-identification, indicating the species, the number of images in each dataset (N), the number of individual identities (c), and providing a link to the dataset.

Ref.	Species	N	c	Notes
Livestock				
[72]	Cattle	8670	181	`https://data.bris.ac.uk/data/dataset/4vnrca7qw1642qlwxjadp87h7`
[9]	Cattle	7043	46	`https://data.bris.ac.uk/data/dataset/10m32xl88x2b61zlkkgz3fml17`

Continued on next page

Ref.	Species	*N*	*c*	Notes
[8]	Cattle	46340	23	`https://data.bris.ac.uk/data/dataset/3owflku95bxsx24643cybxu3qh`
[204]	Yak	2247	103	`Maybeavailableonrequest`
Aquatic Wildlife				
[46]	Humpback Whale	9850	4251	`https://www.kaggle.com/c/humpback-whale-identification`
[87]	Beluga Whale	5902	788	`https://www.lila.science/datasets/beluga-id-2022/`
[3]	Sea Turtle	8729	438	`https://www.kaggle.com/datasets/wildlifedatasets/seaturtleid2022`
[91]	Great White Shark	2456	85	`https://www.saveourseas.com`
[185]	Humpback Whale	7173	3572	`https://www.cascadiaresearch.org/projects/photo-id`
[24]	Killer Whale	86789	367	`https://www.baycetology.org`
[45]	Whale	438613	50271	`https://www.happywhale.com/whaleid`
[185]	Bottlenose Dolphin	10713	401	`https://www.sarasotadolphin.org/meet-dolphins`
[35]	Zebrafish	6672	6	`https://www.kaggle.com/datasets/aalboruniversity/aau-zebrafish-reid/data`
[132]	Saimaa Ringed Seals	2080	57	`https://etsin.fairdata.fi/dataset/22b5191e-f24b-4457-93d3-95797c900fc0`
Terrestrial Wildlife				

Continued on next page

Ref.	Species	N	c	Notes
[68]	Bird	17500	50	https://drive.google.com/drive/folders/1YkH_2DNVBOKMNGxDinJb97y2T8_wRTZz
[125]	Chimpanzee	598	24	"ChimpZoo" https://www.saisbeco.com
[125]	Chimpanzee	1432	71	"ChimpTai" https://www.saisbeco.com
[129]	Giraffe	29806	82	https://www.lila.science/datasets/wni-giraffes
[175]	Panda	6874	50	https://www.github.com/iPandaDateset/iPanda-50
[104, 103]	Elephant	2078	276	https://inf-cv.uni-jena.de/home/research/datasets/elpephants/
[190]	Macaque Monkey	6280	34	https://www.github.com/clwitham/MacaqueFaces
[117]	Zebra	N/A	85	https://www.researchgate.net/publication/221318569_Biometric_animal_databases_from_field_photographs_Identification_of_individual_zebra_in_the_wild
[33]	Gorilla	5428	7	https://vilab.blogs.bristol.ac.uk/2021/01/great-ape-facial-identification/
[120]	Tiger	9496	92	https://cvwc2019.github.io/challenge.html
[138]	Giraffe, Zebra	6925	2056	https://www.lila.science/datasets/great-zebra-giraffe-id

Continued on next page

Ref.	Species	N	c	Notes
Lab Animals				
[130, 149]	(Fruit) Fly	2.6M	60	`https://www.borealisdata.ca/dataset.xhtml?persistentId=doi:10.5683/SP2/JP4WDF`

2.3 Pattern Recognition

Pattern recognition constitutes a fundamental discipline within the domains of artificial intelligence and machine learning, concerned with the automatic identification, analysis, and classification of patterns inherent in data [27]. It encompasses the development of algorithms capable of learning from data, discerning latent structures, and generating accurate predictions or classifications. Central to this field are methodologies such as supervised and unsupervised learning, feature extraction and selection, statistical modelling, and neural network-based techniques, each addressing distinct facets of pattern discovery and representation [27]. The applications of pattern recognition are both extensive and varied, encompassing areas such as biometric identification [93], handwriting recognition, medical image analysis [157], and financial forecasting. By facilitating the processing and interpretation of complex, high-dimensional datasets, pattern recognition underpins the creation of intelligent systems that can adapt to dynamic environments and enhance decision-making processes with improved accuracy and reliability. Moreover, its inherently interdisciplinary character draws upon mathematics, computer science, cognitive science, and engineering, thereby highlighting its importance in both theoretical research and practical technological advancement [27].

2.3.1 Object Classification

Object classification refers to a supervised learning process whereby models are developed and trained to identify and categorise objects within images or videos into predefined classes, utilising techniques from computer vision and machine learning [27]. Training such models necessitates a substantial volume of labelled examples for each class, collectively termed the training set. This dataset facilitates the partitioning of the feature space into distinct regions, each corresponding to a specific class. Once trained, these models should be capable of accurately classifying previously unseen, unlabelled data by determining the region of the feature space to which the object belongs [83].

The performance of a classification model is contingent upon numerous factors. A primary determinant is the quantity and diversity of training data; a larger and more varied dataset enables the model to learn more precise decision boundaries between class distributions, thereby enhancing classification accuracy [61]. Equally significant is the quality of the feature representation: the ability to capture discriminative and representative attributes allows for a clearer separation between classes and thus contributes to improved overall performance. Furthermore, the complexity of the dataset exerts considerable influence; more complex datasets necessitate the division of the feature space into more intricate regions, thereby requiring the employment of more sophisticated models. However, as model complexity increases, so too does the risk of overfitting, wherein the model learns spurious patterns or noise specific to the training data rather than generalisable features [27]. To mitigate this risk, a greater volume of high-quality training data is essential to ensure that the model generalises successfully and performs reliably on previously unseen data.

Significant progress in the field has been driven by the advent of deep learning, particularly through the utilisation of convolutional neural networks (CNNs) [106]. CNNs are capable of automatically learning hierarchical feature representations directly from raw data, thus diminishing dependence on manual feature engineering. These networks can be integrated into larger deep learning architectures that partition the learned feature space into highly nuanced and complex regions, thereby accommodating the demands of intricate and diverse datasets [86]. Their capacity to model sophisticated, non-linear decision boundaries has enabled CNNs to achieve remarkable accuracy and consistency in object classification tasks, even under challenging conditions.

Object classification plays an integral role in numerous real-world applications, including facial recognition systems [162], autonomous vehicles [47], medical imaging [124], and surveillance technologies [123]. In these contexts, accurate classification is essential for enabling decision-making processes, enhancing security protocols, and allowing autonomous systems to interpret and interact successfully with their environment. Nevertheless, challenges such as variations in illumination, occlusion, changes in viewpoint, and intra-class variability necessitate the development of stable and generalisable models [178]. The performance of such models is typically evaluated using metrics such as accuracy, precision, recall, and F1-score, with cross-validation techniques employed to assess their generalisation capability on unseen data.

2.3.2 Object Clustering

Object clustering constitutes an unsupervised learning process in which objects within a dataset are grouped into clusters based on their similarity, typically quantified using a distance metric within the feature space [196]. Smaller distances between objects indicate a higher degree of similarity. In contrast to classification, clustering does not rely on predefined labels or a training set to build a model capable of grouping similar objects. Instead, its primary aim is to uncover the intrinsic structure present within the data, thereby allowing patterns and relationships among objects to emerge autonomously and organically [92, 196].

A wide array of clustering methods exists, reflecting the substantial diversity of real-world data in terms of structure, scale, dimensionality, and underlying distributions [69]. Different approaches have been developed to address these challenges and to accommodate varying notions of similarity and cluster shape [133].

Partitioning methods, such as k-means, assume that clusters are approximately spherical and of similar size [92]. While reliable for simple, well-separated datasets, these methods perform inadequately when confronted with more complex or irregular structures [92]. Hierarchical methods enable a multi-level representation of data relationships without requiring a priori specification of the number of clusters [131]; however, they are often computationally intensive and sensitive to noise [196]. Density-based approaches, such as DBSCAN, are capable of identifying clusters of arbitrary shape and explicitly handling noise, yet they struggle when cluster densities differ significantly [105].

Model-based clustering methods adopt probabilistic assumptions to generate more flexible and interpretable cluster assignments [69]. However, these assumptions regarding data distributions may not always hold in practice. Grid-based approaches, by contrast, prioritise computational efficiency and scalability, making them particularly suitable for large spatial datasets. Nevertheless, they are highly dependent on grid resolution and generally perform poorly in high-dimensional contexts.

The diversity of clustering methods underscores the necessity of accommodating a wide range of data characteristics and analytical objectives. There exists no universally optimal method — a concept formalised by the "No Free Lunch" theorem in machine learning [191] — and thus the selection of an appropriate approach must be guided by the specific properties of the dataset and the goals of the analysis.

Object clustering finds extensive application across various fields, including image segmentation [2], market segmentation, anomaly detection [44], and bioinformatics [142]. In these contexts, it facilitates the discovery of structural groupings and simplifies complex datasets for subsequent analysis. By enabling the autonomous organisation of large volumes of data, clustering plays a vital role in knowledge discovery, providing insights that inform decision-making and shape further analytical processes. Its reliability, however, is heavily contingent on the choice of similarity measures and the quality of feature representation, rendering it an ongoing area of research and refinement.

Constrained Clustering

Constrained clustering constitutes a semi-supervised learning paradigm representing a sophisticated extension of traditional clustering methodologies. Conventional clustering approaches are wholly unsupervised and operate without prior knowledge concerning relationships among data points. Consequently, they may fail to uncover the genuine structure and underlying patterns within a dataset, particularly when these are obscured by complex or high-dimensional feature spaces. By incorporating domain knowledge or user-provided guidance in the form of constraints, the clustering process can be guided towards a more precise and comprehensive representation of the intrinsic data structure [173, 58, 55].

Constraints encode relationships within the data and guide the algorithm towards partitions that respect these relationships. The most commonly employed constraints are instance-level, namely must-link (ML) and cannot-link (CL) constraints. ML constraints require that two instances be assigned to the same cluster, whereas CL constraints stipulate that two instances must not be placed in the same cluster. By integrating such constraints, algorithms produce clustering outcomes that are more consistent with domain knowledge [19].

Constrained clustering has demonstrated considerable success across diverse applications, including object and face clustering in videos [192, 96, 41, 198, 12], document clustering [200], cybersecurity [195], image segmentation [194], and bioinformatics [163]. In document clustering, constraints derived from metadata, citation networks, or expert annotations ensure that semantically related documents are grouped together, enhancing topic modelling and information retrieval [173]. In image analysis, constrained clustering supports more coherent segmentation by enforcing spatial consistency, thereby improving perceptual quality [79].

In bioinformatics, constraints informed by functional relationships or biological pathways facilitate the identification of informative clusters in gene expression or protein interaction data, revealing latent biological processes and improving interpretability [19]. In marketing and e-commerce, constraints can encode demographic or behavioural similarities, enabling more precise customer segmentation and targeted recommendation systems [196]. In anomaly detection, constraints help distinguish normal from anomalous patterns, thereby enhancing fraud detection and cybersecurity measures.

Social network analysis also benefits, as known social ties or group memberships encoded as constraints facilitate accurate community detection and structural analyses [102]. In robotics and autonomous systems, spatial and temporal consistencies serve as constraints to improve object segmentation and environmental understanding, supporting tasks such as navigation and object tracking [127]. In animal ecology and re-identification, spatio-temporal co-occurrence patterns from video or camera trap data enable species-invariant individual tracking, reducing reliance on manual labelling and enhancing system reliability.

Despite its advantages, constrained clustering presents certain challenges, notably the formulation and selection of appropriate constraints, as well as the increased computational complexity associated with enforcing them. The specification of constraints frequently relies on domain experts, who may introduce errors or provide incomplete or suboptimal constraint sets. This dependence also restricts the potential for fully automated clustering workflows, as considerable expert effort may be necessary to develop a sufficiently informative set of constraints [19]. However, in surveillance systems, constraints can be constructed autonomously by exploiting the inherent spatial and temporal relationships among detected objects within video data, thereby reducing reliance on manual specification and enhancing scalability.

Approaches to integrating constraints include modifying the objective function by adding penalty terms that discourage constraint violations, thereby guiding optimisation towards feasible solutions [173, 102]. Alternatively, similarity or distance metrics may be adapted so that ML pairs are treated as highly similar, whereas CL pairs are assigned large dissimilarities [127]. Other methods enforce constraints directly during cluster assignment or adjust the underlying affinity or adjacency matrix in graph-based approaches by strengthening or removing edges. More advanced techniques, such as semi-supervised and probabilistic frameworks, incorporate constraints as priors or probabilistic penalties, enabling a balance between data-driven structures and external knowledge [19].

Online Clustering

Online clustering represents a dynamic extension of traditional clustering methods, specifically devised to process data in a sequential manner by incrementally updating cluster structures as new data points arrive [4, 210]. This approach is particularly well suited to real-time applications and continuous data streams, in which it is neither practical nor feasible to store and repeatedly process the entire dataset.

A central challenge inherent to online clustering is the phenomenon of concept drift, wherein the underlying data distribution evolves over time [71]. Consequently, online clustering algorithms must possess the capacity to adapt to such shifts to ensure that the resulting clusters remain representative as new data is observed.

An additional significant advantage of online clustering is its resource-conscious utilisation of memory. In contrast to batch clustering techniques, which generally require access to the complete historical dataset, online methods retain compact representations or micro-clusters, thereby considerably reducing memory demands [4, 40]. This property is particularly vital for applications involving high-throughput or large-scale data streams, such as real-time video analysis, sensor networks, and online recommendation systems [159, 134].

In addition, online clustering algorithms offer excellent scalability, enabling them to handle continuously growing data volumes without necessitating the reprocessing of prior observations. By incrementally updating model parameters, these approaches can process individual data points with precision and speed, thereby facilitating timely decision-making in streaming contexts [209].

Online clustering approaches can generally be categorised into single-stage and two-stage frameworks. Single-stage methods directly update cluster assignments and model parameters upon the arrival of each new instance, without further refinement. In contrast, two-stage methods, such as CluStream, maintain intermediate summaries (e.g., micro-clusters) during an online phase and periodically perform more detailed clustering of the micro-clusters in an offline phase, thereby striking a balance between adaptability and clustering accuracy [4, 73].

A range of window-based strategies have been developed to regulate the impact of historical data on clustering decisions. The landmark window considers all data from a specific starting point to the present, the sliding window maintains a fixed-size segment that moves forward with time, and the damped window applies a decay function to progressively reduce the influence of older data. These mechanisms enable fine-grained adaptation to dynamic data distributions [71, 210].

Such windowing techniques are central to both single-stage and two-stage online clustering paradigms, providing a principled means of managing the unbounded nature of data streams. In single-stage approaches, clustering is performed directly within the active window—typically a sliding or tumbling window—thereby restricting analysis to the most recent observations and promoting responsiveness to concept drift. Conversely, two-stage methods utilise windowing in the initial, online phase to construct and update micro-clusters, which serve as succinct representations of the data within the current window. These summaries are subsequently leveraged in the offline phase to produce refined macro-clusters. In both cases, window-based strategies support bounded memory usage, computational tractability, and adaptability to temporal variation, making them essential for scalable, real-time stream clustering.

Constraints may also be incorporated into online clustering algorithms, allowing the integration of prior knowledge or domain-specific relationships, even in streaming contexts. By embedding ML and CL constraints into incremental update steps or objective functions, these methods can produce partitions that remain consistent with user-defined relationships while adapting to new data.

Collectively, these advantages render online constrained clustering the most appropriate method for real-time animal re-identification from live video footage.

Cluster Validity Indices

In the context of clustering, their evaluation is typically conducted through the use of cluster validity indices (CVIs), which provide quantitative measures of clustering quality. Such indices may generally be categorised into two principal types: internal and external validity indices.

Internal validity indices assess the quality of the clustering solution based solely on the intrinsic properties of the data, without reference to external information. These indices evaluate aspects such as compactness — the degree to which data points within a cluster are close to one another — and separation — the extent to which different clusters are distinct from each other. Prominent examples include the Silhouette coefficient [146], the Davies–Bouldin index [56], and the Dunn index [62].

By contrast, external validity indices measure clustering performance through comparison with an external ground truth or pre-existing class labels, thereby quantifying the degree to which the discovered clusters correspond to known categories. Notable examples of such indices include the Rand Index and its adjusted version, the ARI [90], NMI [171], and the F-measure.

Together, these evaluation approaches offer a comprehensive framework for assessing the accuracy, stability, and reliability of unsupervised clustering algorithms [196], thereby informing the selection and refinement of methods in practical applications.

Ensemble Methods

Ensemble methods have emerged as a prominent paradigm within machine learning and data analysis, aiming to enhance the performance and reliability of individual models by aggregating the outputs of multiple learners [207]. The fundamental premise underpinning ensemble approaches is that combining diverse models can help mitigate the limitations and biases inherent in any single method, thereby producing more accurate and stable results [116].

In the domain of clustering, ensemble techniques—commonly referred to as cluster ensembles or consensus clustering—operate by integrating multiple clustering solutions into a unified partition that more comprehensively reflects the underlying data structure [168]. These base clusterings may be derived from different algorithms, initialisations, parameter configurations, or subsets of features or data points [202]. By leveraging such diversity, ensemble clustering seeks to reduce sensitivity to noise and initial conditions, both of which are prevalent challenges in conventional clustering methodologies.

The ensemble clustering process is generally divided into two key stages: generation and combination. During the generation phase, a collection of diverse base clusterings is produced. This is followed by the combination phase, in which these results are merged to form a consensus clustering. Various methods have been proposed for this purpose, including co-association matrices, voting-based schemes, and graph partitioning techniques. For instance, the co-association matrix records how frequently pairs of data points are grouped together across different base clusterings, providing a similarity measure for deriving the final partition [94].

A considerable challenge, commonly referred to as the labelling correspondence problem [168] in clustering ensembles, arises from the fact that cluster labels are inherently arbitrary and hold no intrinsic meaning across different clustering solutions. For instance, one algorithm may allocate a particular set of data points to Cluster 1, whilst another may assign the same set to Cluster 3. As these labels serve merely as symbolic identifiers, the direct comparison or aggregation of multiple clustering outcomes can lead to inconsistencies if the label assignments are not appropriately aligned. To mitigate this issue, a range of strategies—such as label alignment [177], permutation matching [7], or the application of label-invariant similarity measures [161]—have been advanced to ensure that clusters across different partitions are compared in a coherent and consistent manner, rather than being obscured by arbitrary labelling conventions.

The advancement of constrained clustering has led to growing interest in incorporating constraints within ensemble clustering frameworks [77]. In particular, semi-supervised clustering algorithms have been employed to generate base partitions for ensemble methods [202], which are subsequently aggregated using various consensus techniques—most commonly majority voting strategies, such as NCuts and CSPA [77].

However, generating base partitions using constrained clustering algorithms can reduce diversity among ensemble members [199]. This is mainly due to the fact that some constrained methods enforce all available constraints strictly, resulting in highly similar or even identical partitions across different runs. Such low variance among ensemble components weakens one of the key advantages of ensemble learning—its capacity to generalise beyond individual clustering outcomes. This challenge may be addressed by using different feature subspaces [202], or by applying a variety of constrained clustering algorithms to obtain heterogeneous partitions [184].

To mitigate this issue, alternative strategies for incorporating constraints into ensemble clustering have been proposed. These include embedding constraints directly into the consensus function [118] and utilising constraint-aware quality metrics to guide the selection of ensemble members [199]. Furthermore, theoretical analysis has demonstrated that, under reasonable assumptions—such as the reliability of prior knowledge and the independence of base clusterings—the accuracy of a semi-supervised clustering ensemble converges asymptotically to 1 as the number of base partitions increases [48].

In summary, ensemble methods represent a flexible and widely adopted approach to improving clustering performance. Their ability to integrate multiple perspectives and incorporate domain knowledge makes them particularly suitable for applications that require high levels of accuracy and interpretability, including image analysis, bioinformatics, and animal re-identification tasks [94].

2.4 Multiple Object Tracking

Multiple Object Tracking (MOT) is a pivotal challenge in computer vision, involving the detection and persistent identification of multiple objects across video frames [52, 121, 14]. Its foundations lie in early radar and surveillance programmes, which predominantly depended on handcrafted motion models and simplistic appearance cues. Initial techniques frequently employed Kalman filters [14] or particle filters [100], placing emphasis on motion continuity at the expense of appearance information. Though successful in controlled or low-density situations, these methods faltered in crowded scenes or during frequent occlusions [201].

Later advances integrated richer appearance models, such as colour histograms and SIFT descriptors, enhancing stability in cluttered environments [174]. The advent of deep learning marked a watershed moment, with convolutional neural networks significantly improving object detection. Consequently, the tracking-by-detection paradigm became predominant in both academic research and real-world applications [52].

This paradigm operates in two principal phases: object detection and object association. First, objects are detected in each frame using modern detectors such as Faster R-CNN, YOLO, or DETR [85, 29, 42]. Next, these detections are associated across frames into trajectories ('tracks') by leveraging appearance characteristics, spatial analysis, and motion dynamics. This approach boasts modularity, enabling independent enhancement of detection and tracking components, integration of state-of-the-art detectors, and scalability to multi-category, large-scale scenarios. Nevertheless, tracking performance remains highly dependent on detection accuracy, and issues such as occlusion and motion blur continue to undermine reliability [52].

Despite notable progress, MOT continues to be hampered by several enduring challenges. Occlusion causes track fragmentation and identity switching [201]. Identity preservation is complicated further when objects share similar appearance traits—common in sports or dense urban areas—leading to frequent mismatches [121, 52]. Re-identification across non-overlapping cameras or long time gaps remains challenging for surveillance and monitoring systems.

Furthermore, achieving a compromise between accuracy and real-time computation is an unresolved issue. Top-performing trackers often rely on elaborate models or global optimisation strategies, which hinder real-world deployment in applications such as autonomous vehicles and live analytics [180]. Domain generalisation also remains limited; models trained under specific conditions often underperform in novel environments featuring different lighting, weather, or background elements [121]. Tracking small or fast-moving objects, such as drones or sports equipment, remains especially difficult due to limited visual information and rapid movement [174].

In recent years, MOT has progressed from traditional motion-based methods to deep learning architectures incorporating transformers, graph neural networks, and end-to-end paradigms [180]. While tracking-by-detection has facilitated significant advances, it continues to face inherent limitations in identity preservation, and real-time performance. Ongoing research focuses on improving accuracy, enhancing efficiency, and developing lightweight, generalisable systems capable of reliable operation in complex real-world environments.

In the context of animal tracking [145, 203, 167, 51], particularly within dynamic and unconstrained environments [100, 137], MOT presents a significant challenge. When animals interact closely—through overlapping movement, physical contact, or social behaviours—standard MOT algorithms often struggle to maintain consistent identity assignment. These difficulties stem from visual ambiguity during interactions, which can cause tracking models to confuse individuals and perform identity switches. Once an identity switch occurs, the error often propagates over time, resulting in random or incorrect identity assignments unless manual correction is performed. This undermines the reliability of long-term tracking and can compromise downstream tasks such as behavioural analysis or movement ecology.

To mitigate these issues, the integration of biometric markers into tracking frameworks has become increasingly important. Biometric features—such as unique coat patterns, facial structures [197], or morphological traits—offer dependable visual cues that persist over time and remain distinguishable even during close interactions. By incorporating these identifiers into MOT pipelines, algorithms can leverage re-identification techniques to reassert the correct identity of each animal following occlusion or interaction events.

This approach, known as re-identification-based tracking, enhances the resilience and stability of MOT systems in complex, multi-animal scenes [197].

Re-Identification-based tracking not only reduces identity switching but also allows for more consistent and long-term monitoring without the need for manual intervention [140]. By matching appearance features extracted from regions of interest (ROIs) to an existing gallery of known individuals, the system can recover from temporary tracking failures and maintain continuity over extended observation periods. This methodology is particularly valuable in ecological studies, wildlife monitoring, and captive animal management, where accurate identity maintenance is critical for understanding individual-level behaviours, health, and social structures.

As such, the integration of animal biometrics into MOT algorithms represents a promising avenue for improving the fidelity and applicability of automated tracking systems [176]. By anchoring identity assignments to species-specific visual signatures, these systems can offer more dependable and scalable solutions for long-term animal monitoring in both natural and controlled environments.

Chapter 3

Benchmark Database

Benchmark datasets play a critical role in the development and evaluation of re-identification systems. They provide a standardised and repeatable framework for comparing algorithm performance across diverse scenarios, enabling the research community to assess progress objectively. In the context of re-identification, where models must accurately recognise individuals or objects across different viewpoints, lighting conditions, and temporal gaps, well-curated datasets are essential for capturing the complexities and variations encountered in real-world applications. Moreover, benchmark datasets often include annotations, identity labels, and challenging conditions that require researchers to identify limitations in existing methods and foster innovation through the development of more stable, generalisable solutions. Without these shared datasets, it would be difficult to ensure fair comparisons or track improvements over time within the field.

Despite advancements in re-identification research, there remains a notable lack of benchmark datasets specifically designed for animal re-identification. Unlike human re-identification, which benefits from a wealth of large-scale, annotated datasets, the animal domain suffers from limited, fragmented, and often species-specific datasets that lack consistency in data quality, annotation standards, and scale. This scarcity hampers the development and fair evaluation of algorithms intended for wildlife monitoring, conservation efforts, and behavioural studies. The absence of comprehensive, standardised benchmarks makes it difficult to assess generalisability across species or environments, impeding progress in building stable, transferable models for real-world animal tracking scenarios. Addressing this gap is essential to advancing automated animal identification technologies and their deployment in ecological and zoological research.

This chapter presents an overview of our benchmark dataset, accompanied by an in-depth analysis of the challenges it poses. It also includes an experimental study aimed at determining which feature representation performs best when a range of classification methods are applied. The highest-performing feature representation identified in this study will be used in all subsequent experiments.

In an unsupervised re-identification context, the sole information available for processing comprises unlabelled bounding box detections produced by a multi-object tracking algorithm. The subsequent objective is thus to aggregate and assign these unlabelled detections to distinct identity groups, whilst addressing the inherent complexities and ambiguities present within the data. It is therefore essential to explicitly identify and articulate the challenges posed by such data, in order to inform the development of successful solutions. Moreover, it is imperative to employ the most informative and discriminative feature representations to support this task.

Contributions covered in this Chapter

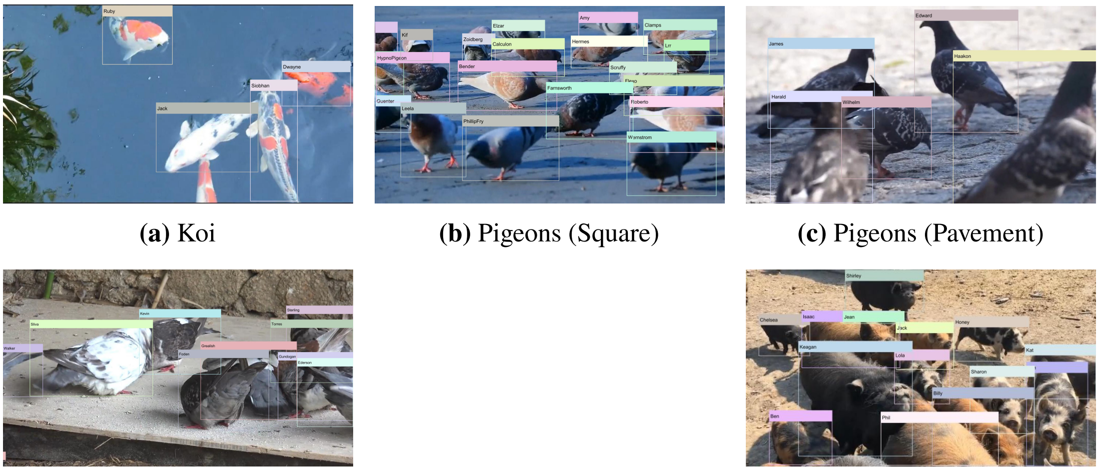
Creating a 5-video dataset including the Ground Truth (GT) annotations. Providing an in depth analysis of the characteristics and challenges involved.

Publications 1 & 2 in Section 1.4

1

3.1 Overview

This dataset can be found at `https://zenodo.org/records/7322821`, and comprises five short videos sourced from Pixabay `https://pixabay.com/` under the Pixabay license, with durations ranging from 9 to 24 seconds. Each video features a group of animals belonging to a single species, capturing their natural movements within the scene. While most animals remain visible throughout the clips, some intermittently enter and exit the camera's field of view. All videos were manually annotated with individual animal identities, and the annotations are provided in a standardised format across the dataset. Representative frames from each video are shown in Figure 3.1, and the characteristics of each video are summarised in Table 3.1.



(a) Koi

(b) Pigeons (Square)

(c) Pigeons (Pavement)

(d) Pigeons (Kerb)

(e) Pigs

Figure 3.1: Illustrative examples of annotated frames from the animal re-identification database. Each animal is enclosed within a bounding box and labelled with its corresponding identity, which serves as the ground truth for the subsequent evaluation of the proposed methods

Table 3.1: Characteristics of the videos

Video	T	L	N	c	Min p/f	Max p/f	Avr p/f	Imbalance
Koi fish	536	22	1635	9	1	6	3.1	2.8
Pigeons (square)	300	9	4892	27	1	23	16.3	24.8
Pigeons (pavement)	600	24	3079	17	3	8	5.1	19.3
Pigeons (Kerb)	443	17	4700	14	8	13	10.6	3.1
Pigs	500	16	6184	26	4	20	12.4	10.5

Table notes: T is the number of frames; L is the video length in seconds; N is the number of objects (individual animal clips); c is the number of classes (animal identities); Min p/f is the minimum number of animals per frame (image); Max p/f and Avr p/f are respectively the maximum and the average numbers; Imbalance represents the size of the largest class divided by the size of the smallest class.

Each dataset comprises a collection of BBs, parametrised by the top-left corner coordinates (x, y) and the spatial extent defined by width w and height h. These are encoded as a 4-tuple (x, y, w, h) and are linked to the corresponding video frame index and a distinct identity label assigned to the tracked object. Additionally, each dataset is provided with 5 distinct feature representations, each capturing unique characteristics that can be leveraged to differentiate between objects.

3.2 Feature Extraction

Designing a species-invariant re-identification solution presents a range of challenges and design constraints. Most existing online approaches are species-specific, and are therefore considerably less complex than those intended to operate across multiple species. While deep learning has become foundational in modern machine learning, artificial intelligence, and pattern recognition—often delivering superior performance—its effectiveness is typically maximised in single-species scenarios. In such contexts, deep neural networks can extract highly specialised and distinctive feature representations for individual animals, resulting in well-separated clusters that enhance the performance of downstream clustering algorithms.

However, due to the substantial inter-species variability in biometric traits, applying deep learning for feature extraction across multiple species is largely impractical as the amount of available training data is insufficient to produce robust representations capable of reliably distinguishing individuals. Despite this limitation, we have included certain deep feature representations within this benchmark dataset. The extracted feature types for each of the five datasets are as follows:

- AE (*Autoencoder*). An autoencoder is a specialised deep learning architecture designed to encode input data into a lower-dimensional latent space, followed by reconstruction through a decoding process that mirrors the encoding procedure, as illustrated in Figure 3.2. By omitting the decoder, the encoder can be employed as a feature extractor, enabling dimensionality reduction by projecting the data into a latent representation learned via deep learning.

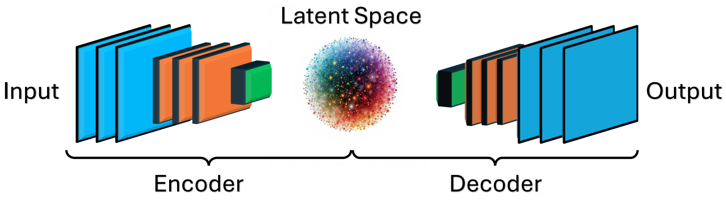


Figure 3.2: Depiction of the general architecture of an autoencoder network. The illustration demonstrates how the input is processed through the encoder component and mapped to a latent space representation, which is subsequently reconstructed into the original input via the decoding process.

To extract the AE features for each dataset, we utilised the MATLAB function `trainAutoencoder` with its default parameter settings to establish a baseline representation without manual tuning, ensuring consistency and reproducibility across all datasets. This produced a latent representation of size 10, yielding 10 AE features. The network was trained on the complete dataset.

- HOG (*Histogram of Oriented Gradients*). HOG features are extensively used in computer vision for object detection [156, 181, 97], as they encode the structural attributes of an image by analysing gradient orientations and magnitudes within localised regions known as cells. This approach is especially effective at characterising shapes and edges, making it well-suited for object detection tasks and potentially applicable for distinguishing individuals within video sequences.

  Each image was resized to a square by extending the image outward from its central point until one of the edges, either the height or the width, was reached. The HOG features were then extracted from the colour image using MATLAB's `extractHOGfeatures` function with its default settings to create a baseline representation. The resulting feature vector comprised 576 HOG features.

- LBP (*Local Binary Patterns*). Local Binary Patterns (LBP) is a texture descriptor widely utilised in image processing and computer vision to capture the local structural features of an image. It works by analysing the neighbourhood of each pixel—excluding those on the borders—and comparing the intensity values of surrounding pixels with that of the centre pixel. A new value is then assigned to the centre pixel based on this comparison. Once the entire image has been processed, the resulting LBP pixel values are typically converted into a histogram, which can be used for classification and recognition tasks.

  To extract the LBP features, each image was resized and converted to greyscale, followed by the application of the MATLAB function `extractLBPfeatures` with the default parameters, with the exception of the 'Upright' setting, which was set to false to allow for rotationally invariant features. The resulting feature representation comprised 10 LBP features.

- MN2 (*MobileNetV2*). For the MN2 features, we employed the Keras MobileNetV2 model pre-trained on ImageNet; the network configuration is illustrated in Figure 3.3. By removing the final layer and replacing it with a GlobalAveragePooling layer, we obtained a feature representation with a dimensionality of 1280. MobileNetV2 was selected due to its lightweight architecture and strong performance on a range of visual recognition tasks, making it well-suited for efficient feature extraction.

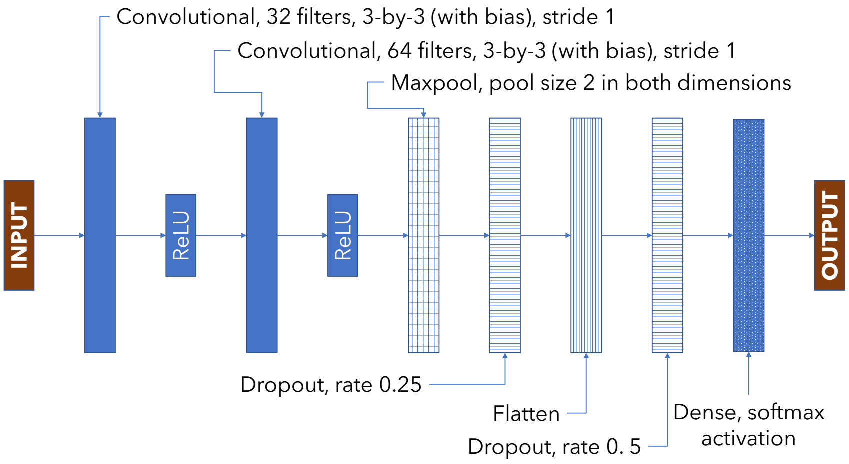

Convolutional, 32 filters, 3-by-3 (with bias), stride 1
Convolutional, 64 filters, 3-by-3 (with bias), stride 1
Maxpool, pool size 2 in both dimensions
INPUT
ReLU
ReLU
OUTPUT
Dropout, rate 0.25
Flatten
Dropout, rate 0. 5
Dense, softmax activation

Figure 3.3: Visualises the MobileNetV2 network architecture, showing the network's layers along with the corresponding activation functions and layer parameters.

- RGB (*RGB Moments*). RGB moments describe the colour features present within an image. They are extracted using the following process: the image—or, in this case, the detection—is divided into 3-by-3 blocks of equal size. Each of the colour channels (red, green, and blue) is then separated. For each block, the mean and standard deviation are calculated for each of the three colour channels. These values are then stored, resulting in a 54-dimensional feature vector. An illustrated example of the process is depicted in Figure 3.4.

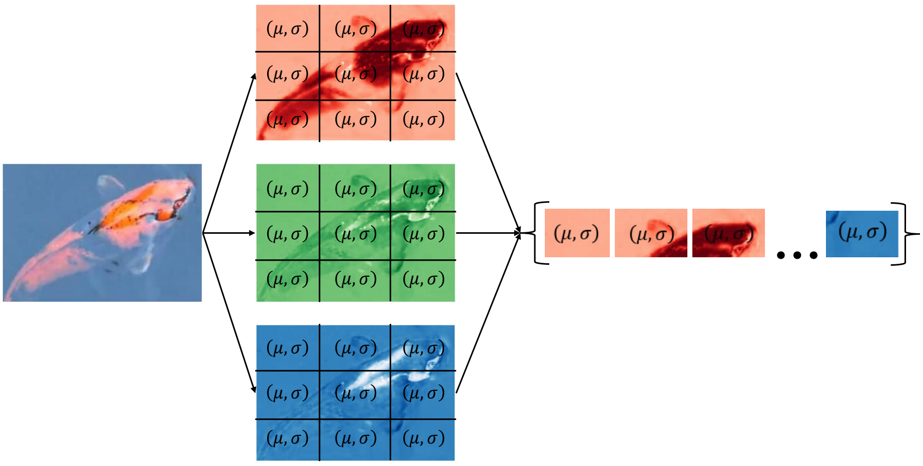


Figure 3.4: Illustrative example of the RGB feature extraction process, showing the separation of the colour planes and the subsequent computation of the mean and standard deviation values for each block.

It should be noted that the development of each feature representation is based on distinct extraction methodologies, which consequently result in notable differences in dimensionality. Such variation can render certain representations less suitable for clustering tasks due to the curse of dimensionality—a concept that encompasses the challenges inherent in analysing data within high-dimensional spaces, where the number of features far exceeds the capacity of standard algorithms to process effectively. With increasing dimensionality, the feature space grows exponentially, causing data to become sparse and undermining the reliability of statistical inference. In these conditions, traditional measures of distance and similarity lose their discriminative power, as data points tend to appear uniformly distant from one another, thereby weakening the performance of methods such as clustering, nearest-neighbour search, and density estimation. Additionally, the computational burden of learning tasks escalates with dimensionality, frequently leading to overfitting and reduced generalisability.

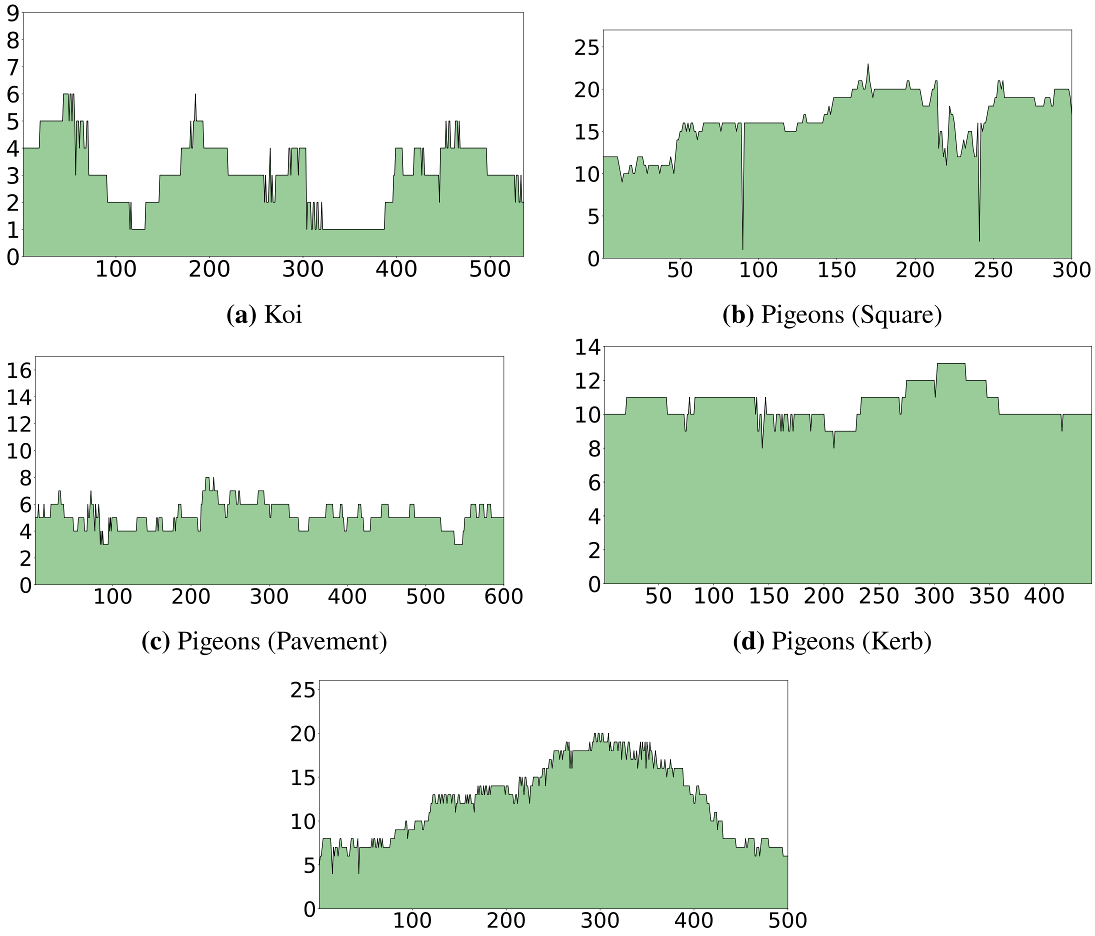
3.3 Dataset Challenges

There are many challenges involved in processing video footage, largely due to the unpredictable nature of the subjects within the frame. As these challenges cannot be controlled at the source, any effective video analysis solution must incorporate methods to address them. Some of the most common challenges found in surveillance video footage—many of which are also present in our datasets—are highlighted in this section.

3.3.1 Multiple Objects

The number of objects within a single frame significantly influences the complexity of developing a solution to the proposed problem. As the number of objects increases, so does the difficulty, primarily due to higher levels of occlusion. This added complexity affects object detection and tracking, while the likelihood of feature blending—particularly in dense scenes—rises. Such conditions make the task of re-identification considerably more challenging, as feature representations may become diluted or contaminated.

Figure 3.5 illustrates the number of objects present in each frame across the videos in the dataset, in relation to the number of classes represented. It is evident that the number of objects per frame varies throughout each video; however, there are consistently multiple objects within each frame, with the exception of a few outlier frames observed in the Pigeons (Square) video and a handful of frames in the Koi dataset.



(a) Koi

(b) Pigeons (Square)

(c) Pigeons (Pavement)

(d) Pigeons (Kerb)

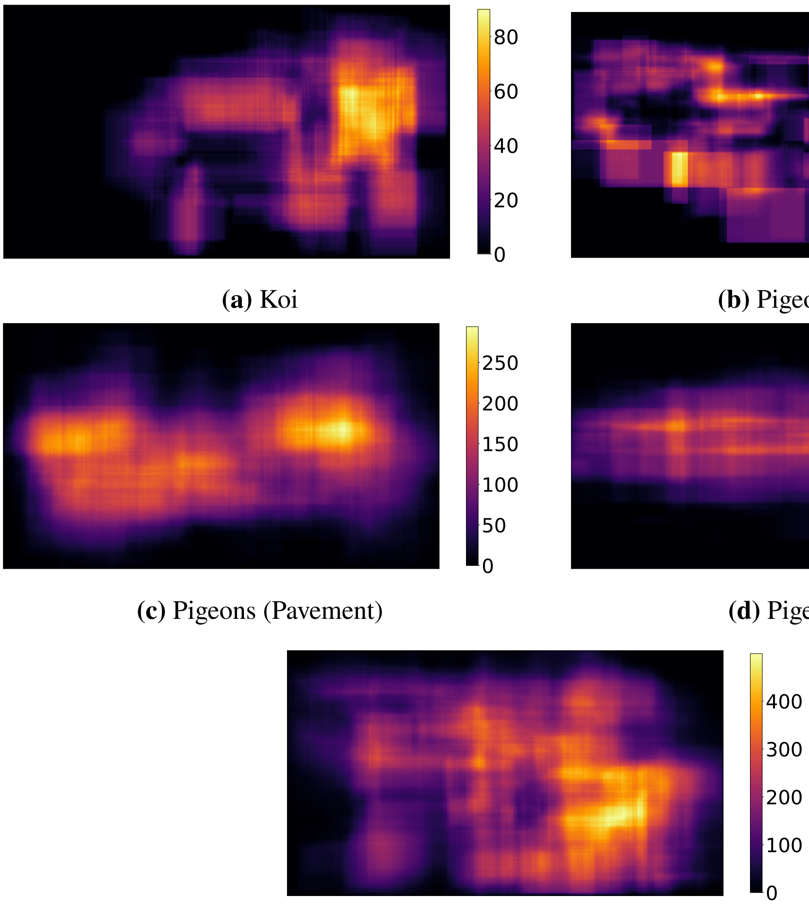
(e) Pigs

Figure 3.5: Visualises the number of objects per frame across the five datasets. Higher numbers of objects within a single frame correspond to increased complexity of the task.

3.3.2 Occlusion

Occlusion refers to the extent to which a subject within a frame is blocked or obscured by another object. Separating an occluded subject from the rest of the scene is extremely challenging without depth perception and must instead be handled through alternative techniques. As feature extraction methods are designed to process BBs with a consistent number of descriptors, the content within each bounding box plays a crucial role in accurately representing the subject. If occlusion occurs within a bounding box, the resulting feature description may inadvertently include elements of another identity, contaminating the representation. This section highlights the levels of occlusion present in our dataset and illustrates the difficulty of developing solutions capable of managing this challenge.

To visualise the location and frequency of occlusion within each video, occlusion heatmaps are presented in Figure 3.6. For each video, a matrix matching the dimensions of the video frame is initialised, with all values set to zero. Each value represents the number of times a corresponding pixel in the image has been occluded. For every frame in each video, the intersection between all pairs of BBs is computed, and every pixel within these intersecting regions is incremented by one. After processing all frames, the resulting matrix reveals both the spatial distribution of occlusions and the number of times each pixel has been affected.



(a) Koi

(b) Pigeons (Square)

(c) Pigeons (Pavement)

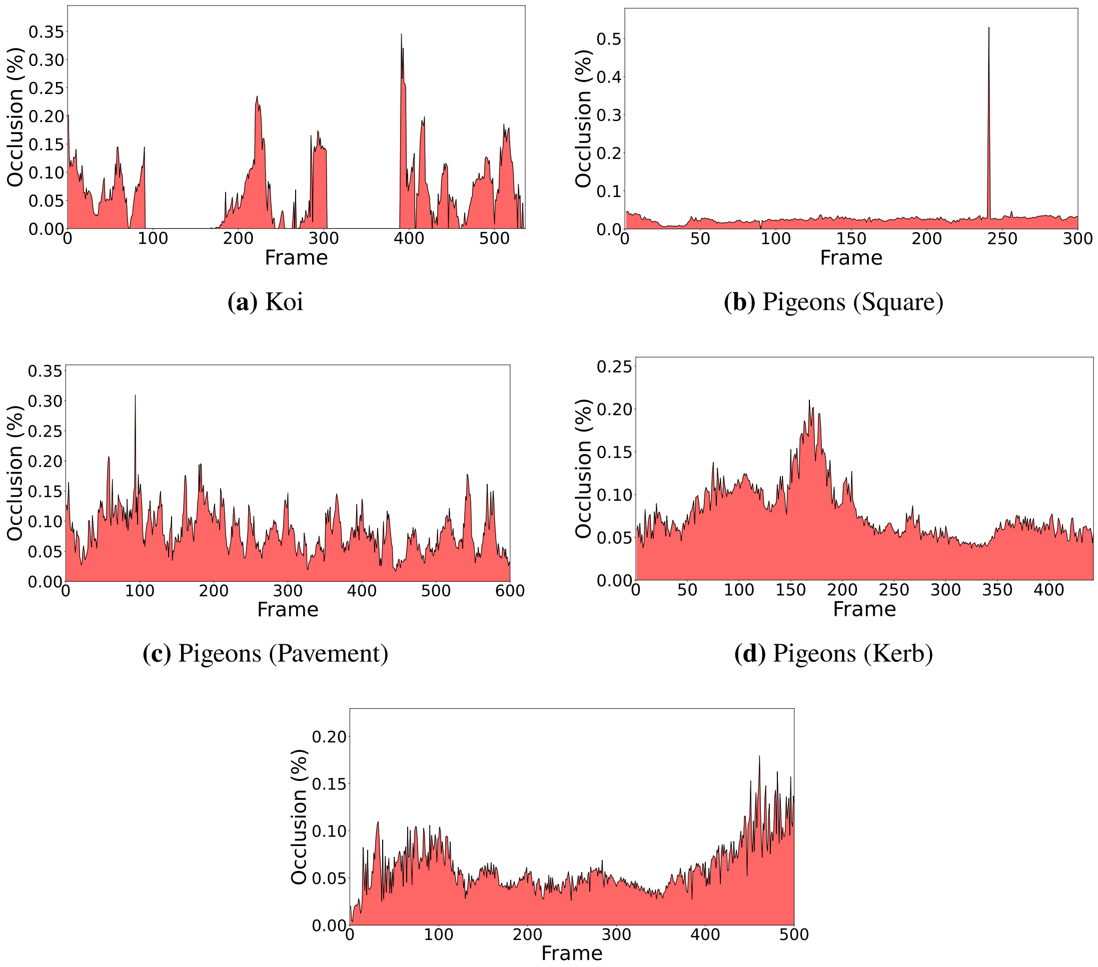
(d) Pigeons (Kerb)

(e) Pigs

Figure 3.6: Displays occlusion heatmaps for each of the five datasets, with brighter regions indicating a higher frequency and spatial concentration of occluded objects within the frame, while the colour bar represents the relationship between the number of occlusions and the plot intensity.

The visualisations in Figure 3.6 show that occlusion within the videos is not confined to a specific area or small section of the frame. As objects are free to move throughout the scene, any identity may experience occlusion at various points, potentially leading to its feature representation being contaminated by multiple other identities present in the video.

While the location and frequency of occlusion provide valuable insight into the presence of occlusion, they do not offer a complete understanding of its impact within the video. Another crucial factor is the proportion of each detection that is actually obstructed by another object. This percentage can indicate the extent to which a feature representation may be contaminated—higher occlusion percentages imply greater contamination, as more of another identity may be included in the representation. To visualise these levels, the average percentage of occlusion is calculated for each pair of BBs detected in every frame. This is done by computing the intersection area between a pair of BBs and dividing it by the area of the bounding box of the individual detection, thereby estimating the proportion of the object that is occluded. The average occlusion value is then calculated for each frame, the results of which are shown in Figure 3.7.



(a) Koi

(b) Pigeons (Square)

(c) Pigeons (Pavement)

(d) Pigeons (Kerb)

(e) Pigs

Figure 3.7: Displays the average percentage of object occlusion within each frame of each video, where each value represents the mean proportion of an object occluded by others, averaged across all objects in the frame.

Figure 3.7 presents the average percentage of object occlusion per frame across the five video sequences. The plots indicate that the extent of occlusion varies considerably from one frame to another. Nevertheless, with the exception of the Koi video, all other sequences demonstrate a consistent presence of occlusion throughout. The anomaly observed in plot (b) arises due to the presence of only two objects within the frame, whose corresponding BBs exhibit substantial overlap, thereby resulting in a high percentage of occlusion for that particular frame.

Table 3.2: Occlusion percentage values averages across frames for each datasets.

Dataset	Overall Occlusion (%)
Koi	5.09
Pigeon (Square)	2.68
Pigeon (Pavement)	8.48
Pigeon (Kurb)	7.93
Pigs	6.09

Table 3.2 presents the overall occlusion for each dataset. Each value was derived by computing the average occlusion across all frames within each video, thereby yielding a representative measure of the occlusion present throughout the video.

3.3.3 Concept drift

Concept drift refers to the phenomenon where the distribution of an identity within the feature space changes over time. When the drift is significant, it can render pre-trained models ineffective at correctly assigning class labels. Concept drift is primarily a concern in online processing and real-time applications, and as such, it will only be considered when developing solutions intended for those scenarios. Nonetheless, assessing the degree of concept drift present in each dataset is important, as it highlights the complexity involved in creating a dependable and adaptable solution.

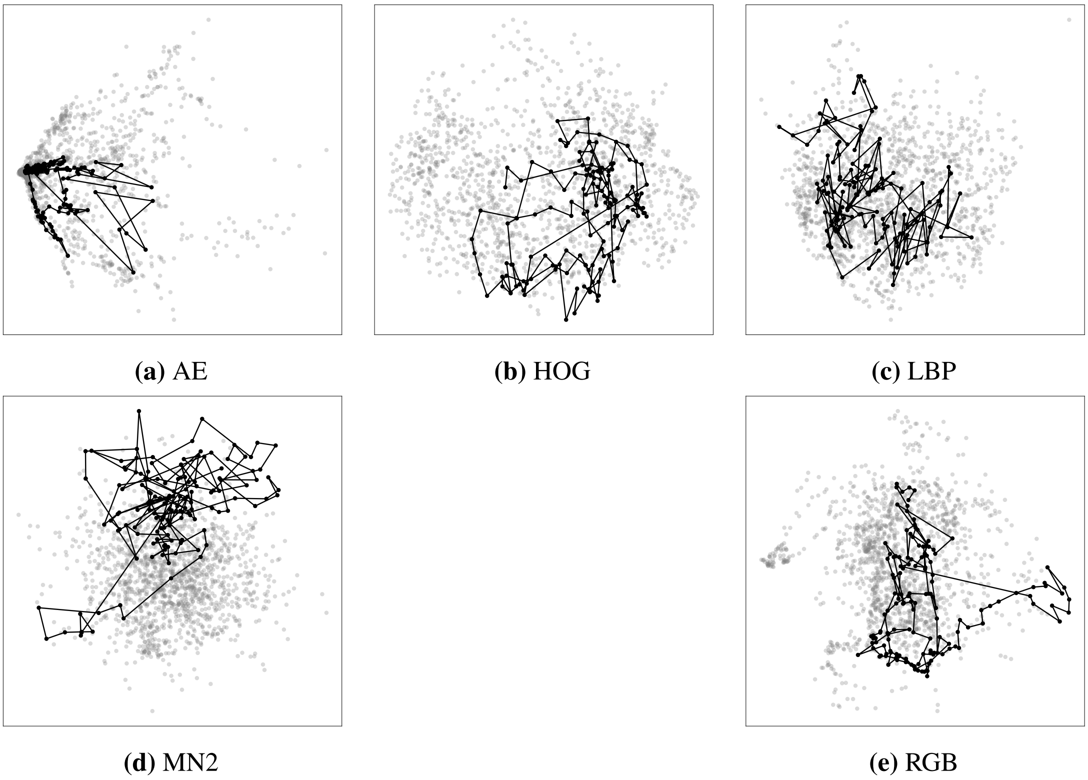
Figures 3.8, 3.9, 3.10, 3.11, and 3.12 present each dataset in the space of the first two principal components (PCA)[88], for the five feature representations. It is important to note that differences in dimensionality between the feature representations and any subsequent dimensionality reduction for visualisation, as in this case, can produce substantially different projections. In datasets with a very high number of dimensions, such as MN2, the curse of dimensionality and the consequent convergence of distances between points may hinder a feature reduction algorithm, such as PCA, from identifying the features that best capture the maximal variance in the dataset. As a result, intrinsic structures within the data may be obscured or lost after projection to a visible number of dimensions.

In each visualisation, a single, consistent identity is highlighted in black within the feature space, with temporal progression represented by lines connecting each point to its immediate successor. All other identities visible within the video are shown in light grey. This visual distinction enables an intuitive assessment of the focal identity's dispersion relative to the broader feature distribution. While not a quantitative metric, this approach offers visual insight into the extent of variation that an identity's representation may exhibit over time.

The aforementioned figures demonstrate that the feature representation of a single identity can spread across a substantial portion of the feature space over time, irrespective of the representation method employed. This dispersion highlights the variability arising across frames, suggesting that an individual's representation is not confined to a tightly bounded cluster but instead spans a broader region. Such behaviour reveals temporal fluctuations in the extracted features and offers clear evidence of concept drift throughout the progression of a video.

One of the contributors to concept drift observed within the datasets is the intermittent presence of animals within the video frames. When an animal exits and subsequently re-enters the frame, it may do so in a previously unseen orientation, potentially revealing aspects of its appearance that were not previously captured. This can result in a novel feature representation that differs significantly from those previously associated with the same identity, thereby introducing substantial variability into the feature space.

Figure 3.13 depicts a single identity from the pig video using the RGB feature representation, following dimensionality reduction via PCA for visualisation purposes. The identity's trajectory is segmented into continuous periods of presence, each represented by a distinct colour. Whenever the animal exits and subsequently re-enters the scene, a new colour is assigned to its trajectory. The sequence of these appearances, along with their corresponding colours, is indicated in the plot legend, where 0 denotes the animal's initial appearance and track 7 its final one. This colour segmentation enables clear visual comparison between different episodes of presence, illustrating how



(**a**) AE

(**b**) HOG

(**c**) LBP

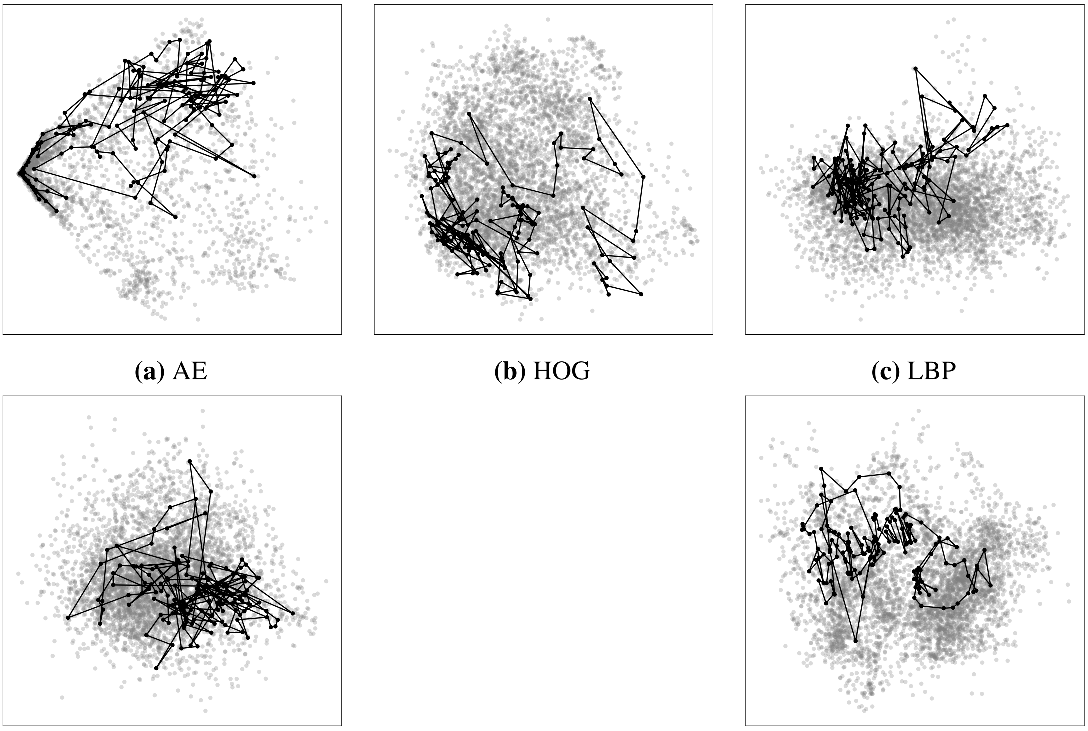
(**d**) MN2

(**e**) RGB

Figure 3.8: 2D representation of the Koi dataset using PCA for dimensionality reduction applied to all feature representations. A single identity, consistent across all plots, is highlighted in black, visualising the temporal evolution of its feature representation relative to the overall feature space, shown in light grey.

the feature representation evolves between reappearances. Notably, most reappearances occupy distinct regions of the feature space relative to earlier tracks. As in previous figures, all other identities within the video are shown in grey, providing context for the focal identity's dispersion within the overall feature distribution.

While Figure 3.13 illustrates the evolution of the feature representation across multiple reappearances, it also reveals how the representation changes within a single period of continuous presence—depicted by a single colour in the plot. This intra-window evolution reflects the degree of concept drift occurring even within a single appearance. Notably, the extent of this drift tends to increase with the duration of the animal's presence in the frame. This can be attributed to the fact that longer visibility allows for a greater variety of the animal's features to be observed and, consequently, incorporated into the feature representation. Additionally, extended periods in the frame increase the likelihood of occlusion events, which may introduce noise or distortion into the feature data, further contributing to concept drift within a single appearance window.



(a) AE

(b) HOG

(c) LBP

(d) MN2

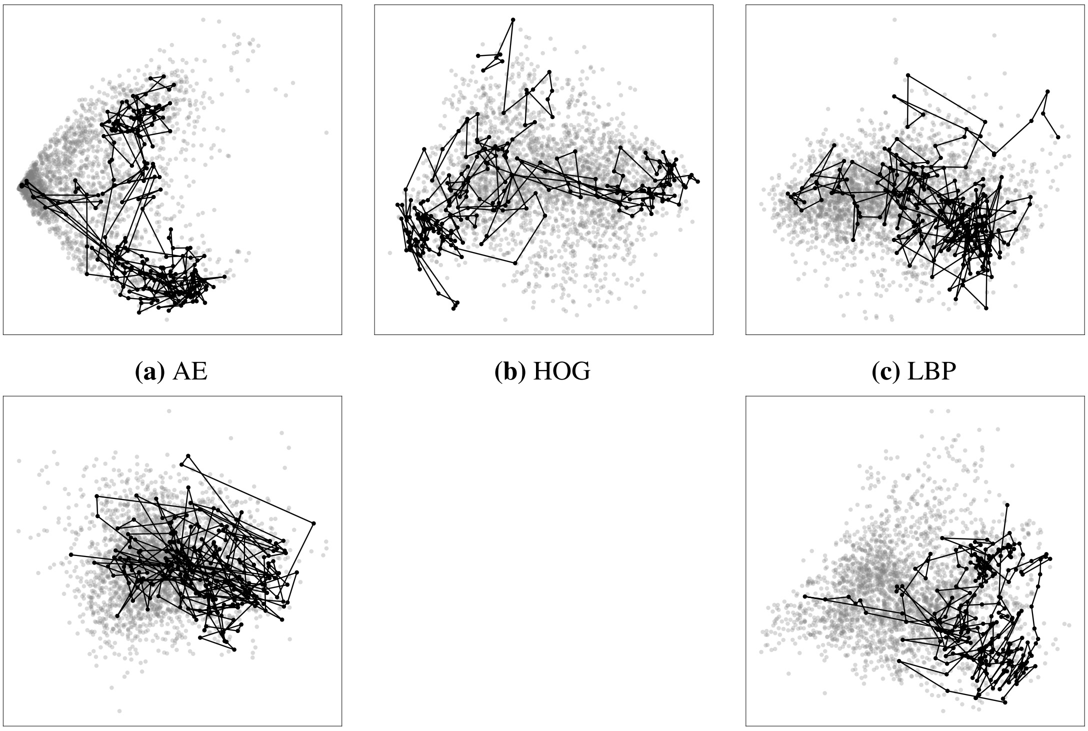
(e) RGB

Figure 3.9: 2D representation of the Pigeon (Square) dataset using PCA for dimensionality reduction applied to all feature representations. A single identity, consistent across all plots, is highlighted in black, visualising the temporal evolution of its feature representation relative to the overall feature space, shown in light grey.

Although numerous metrics exist to quantify the extent of concept drift within data [59, 109, 183, 182, 141], their applicability depends heavily on the nature of the data being analysed. Each metric requires careful consideration to ensure it aligns with the characteristics and assumptions of the dataset. Consequently, I have chosen not to incorporate a specific calculated metric to quantify concept drift in this work. Instead, I rely on the visual examples presented in this section, which effectively illustrate the presence and nature of concept drift within the datasets. These visualisations provide an intuitive understanding of how feature representations evolve over time—both within individual appearances, across successive reappearances, and throughout the entire duration of each video.

3.3.4 Intra-Cluster Compactness Vs Inter-Cluster Separability

Inter-cluster separability fundamentally influences the difficulty of classification and clustering tasks. The ability to distinguish between different clusters forms the foundation of performance in these tasks. Consequently, when clusters exhibit high similarity, it becomes increasingly challenging to separate them effectively. This issue is particularly



(a) AE

(b) HOG

(c) LBP

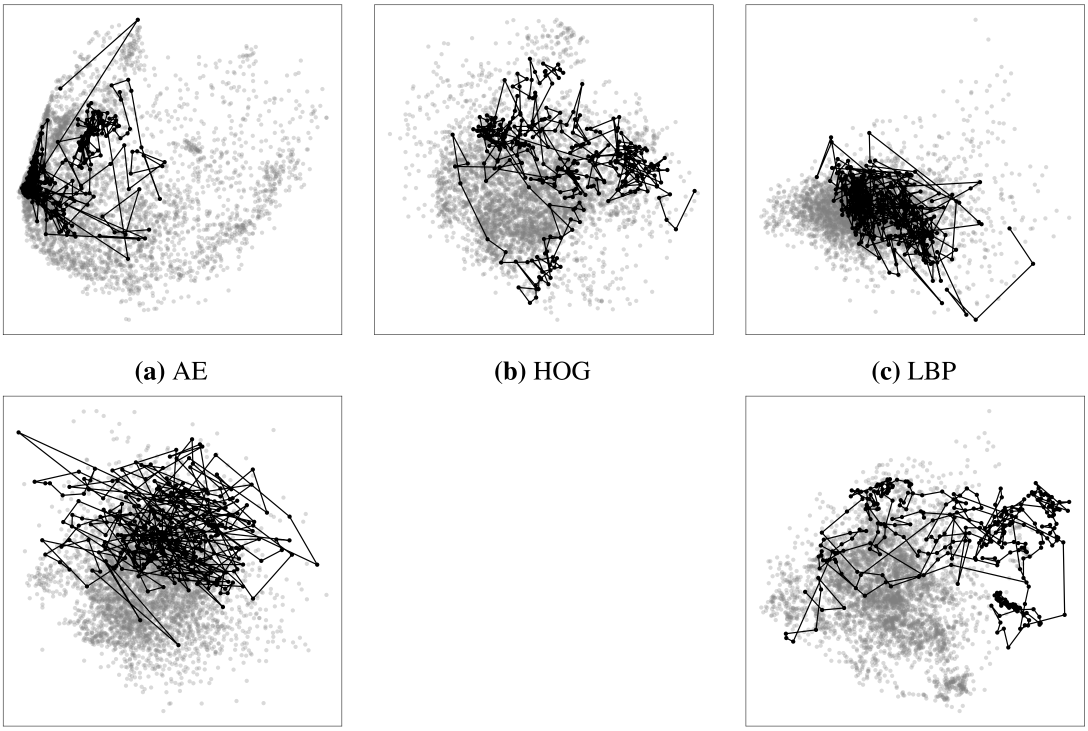
(d) MN2

(e) RGB

Figure 3.10: 2D representation of the Pigeon (Pavement) dataset using PCA for dimensionality reduction applied to all feature representations. A single identity, consistent across all plots, is highlighted in black, visualising the temporal evolution of its feature representation relative to the overall feature space, shown in light grey.

pronounced in our animal datasets, where each video contains multiple individuals of the same species. These individuals are often visually indistinguishable even to the human eye, necessitating reliance on additional information—such as the capacity to track an object over time—to differentiate between nearly identical animals. To visualise the similarity between identities within the datasets, Figure 3.14 presents the mean appearance of each identity across the entire duration of their respective videos. These figures highlight the low inter-cluster variance and visually emphasise the challenge of distinguishing between identities, as many appear nearly identical even to the human eye.

While Figure 3.14 visually demonstrates the complexity involved in distinguishing between identities—and, by extension, the challenge inherent in their corresponding feature representations—it does not quantify the complexity of the structures within each feature space. This structural complexity ultimately dictates the difficulty of the clustering problem under consideration.



(a) AE

(b) HOG

(c) LBP

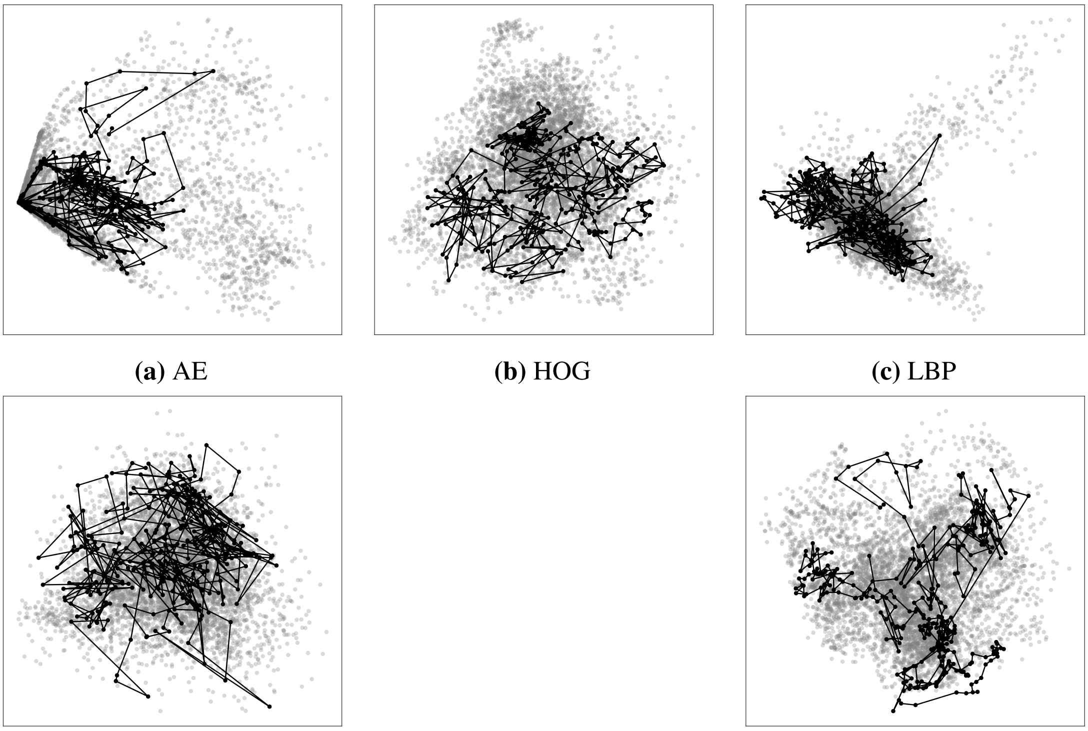
(d) MN2

(e) RGB

Figure 3.11: 2D representation of the Pigeon (Kerb) dataset using PCA for dimensionality reduction applied to all feature representations. A single identity, consistent across all plots, is highlighted in black, visualising the temporal evolution of its feature representation relative to the overall feature space, shown in light grey.

CVIs are commonly used to evaluate the quality of clustering results [13, 84]. Internal CVIs assess clustering performance based solely on the data and the resulting cluster structure, while external CVIs require ground truth labels and compare the predicted clusters to the actual labels. To quantify how challenging our datasets are to cluster, we can take advantage of the design characteristics of internal CVIs.

Internal CVIs are constructed based on a combination of intra-cluster compactness and inter-cluster separation [101], under the principle that clustering quality improves when clusters are both tightly grouped and well-separated from one another. By applying an internal CVIs to our datasets—using the ground truth labels in place of predicted clusters—we can estimate how inherently difficult the datasets are to cluster. Poor CVI scores would suggest low compactness within clusters and poor separation between them, indicating that the true structure of the data is complex or poorly defined, and therefore harder to recover through clustering.



(a) AE

(b) HOG

(c) LBP

(d) MN2

(e) RGB

Figure 3.12: 2D representation of the Pigs dataset using PCA for dimensionality reduction applied to all feature representations. A single identity, consistent across all plots, is highlighted in black, visualising the temporal evolution of its feature representation relative to the overall feature space, shown in light grey.

The Silhouette Index [146] was selected as the internal clustering validation index for this purpose due to its point-level evaluation approach rather than utilising aggregate cluster-level measures such as centroids, as is done in the Davies–Bouldin index [56]. Intra-cluster compactness is assessed based on how close each point is to other points within the same cluster, while inter-cluster separation is measured by the distance from that point to points in the nearest neighbouring cluster. Additionally, it is one of the few internal CVIs that is inherently normalised in the range $[-1, 1]$, making it particularly well-suited for comparing clustering solutions or, in this case, different feature representations.

The Silhouette Index is a summation-based metric, where higher values indicate better clustering quality or more easily separable data. The index for a clustering solution or partition $\mathcal{P}$ is calculated using equations 3.1-3.3:

$$Sil(\mathcal{P}) = \frac{1}{N} \sum_{C_k \in C} \sum_{x_i \in C_k} \frac{b(x_i, C_k) - a(x_i, C_k)}{\max\{a(x_i, C_k), b(x_i, C_k)\}}, \tag{3.1}$$

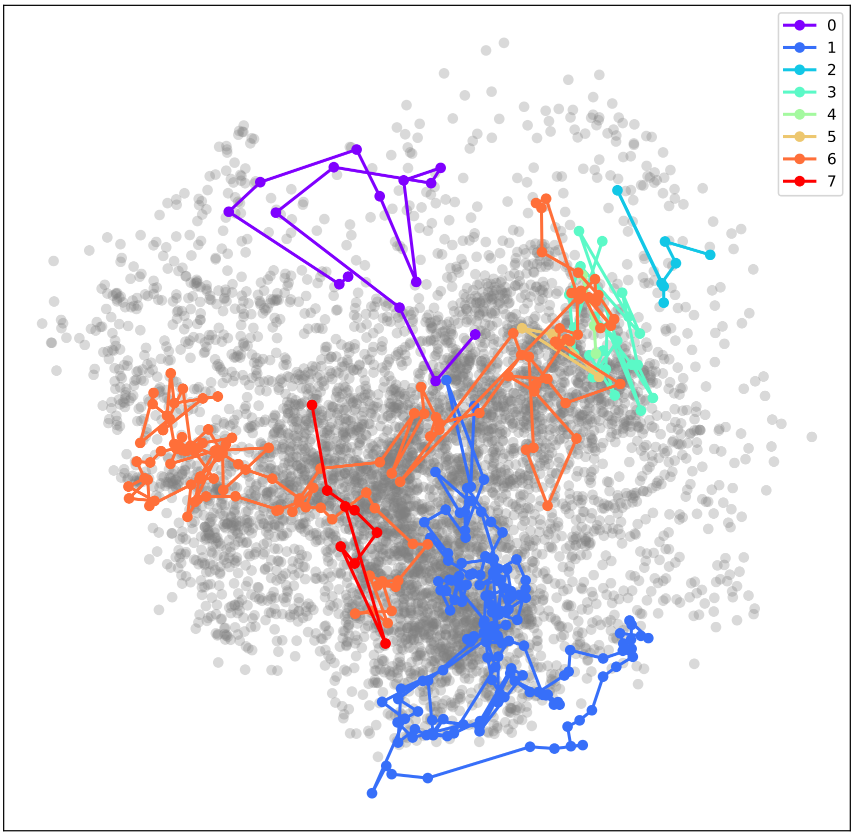


Figure 3.13: PCA visualisation of the tracked reappearances of a single identity in the Pigs dataset, using the RGB feature representation. Each appearance of the identity is numbered and depicted in a distinct colour, as indicated by the legend.

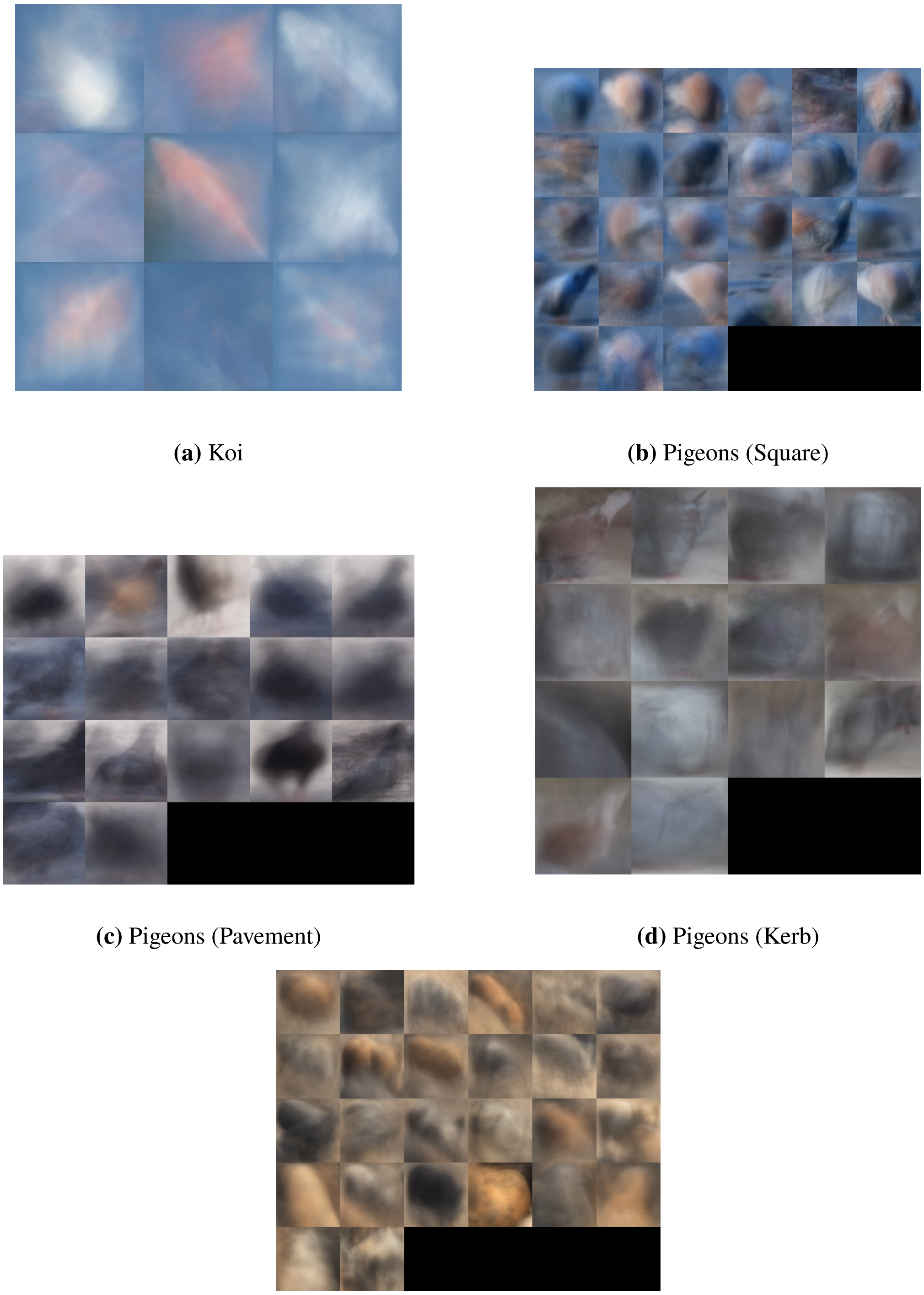
where N is the number of data points, $C = \{C_1, \ldots, C_K\}$ is the set of K clusters, x_i is a point in the data set, and

$$a(x_i, C_k) = \frac{1}{|C_k|} \sum_{x_j \in C_k} d_e(x_i, x_j) \tag{3.2}$$

$$b(x_i, C_k) = \min_{C_l \in C \setminus C_k} \left\{ \frac{1}{|C_l|} \sum_{x_j \in C_l} d_e(x_i, x_j) \right\} \tag{3.3}$$

Here $|\cdot|$ denotes cardinality, and $d_e(x_i, x_j)$ is a chosen distance between objects x_i and x_j. In this case, $d_e(x_i, x_j)$ was defined as the Euclidean distance.

Table 3.3 presents the silhouette scores for each dataset across five distinct feature representations. These scores were computed using the ground-truth labels, thereby providing insight into the underlying structural complexity of the data. The table offers



(a) Koi

(b) Pigeons (Square)

(c) Pigeons (Pavement)

(d) Pigeons (Kerb)

(e) Pigs

Figure 3.14: An example of inter-class similarity. In each subplot (corresponding to a video), the BBs for every identity have been averaged into a single representation—depicting their mean appearance across all frames in which they are present.

an indication of which feature representations may be more appropriate for clustering tasks. Representations yielding the lowest scores are highlighted in red, while those achieving the highest scores are shown in green. The LBP features appear to be the least effective, consistently producing the lowest silhouette scores.

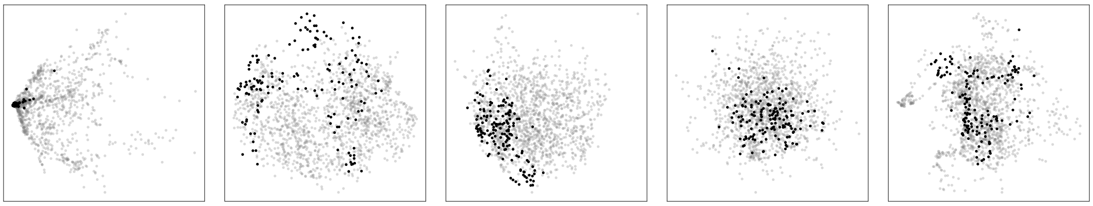
All values in the table are negative, reflecting the poor separability between identities. This suggests that the clusters within the data are highly intertwined, making them challenging to distinguish through clustering. Furthermore, feature representations with a higher number of dimensions tend to yield higher silhouette values, indicating improved separability between identities. This observation highlights the inherent spatial sparsity of high-dimensional data and its influence on cluster separability. However, as noted previously, increasing dimensionality also exacerbates the difficulty of clustering due to the convergence of distances between points in high-dimensional space. Consequently, the most favourable feature representation is one with sufficiently high dimensionality to enhance cluster separability while retaining the discriminative power of distance and similarity metrics for subsequent clustering tasks. In this context, RGB features appear to meet these criteria most effectively. We observe that this metric alone does not definitively determine the most suitable feature representation for clustering. A classification experiment would be more suitable for selecting among the feature representations.

Table 3.3: Silhouette scores for each dataset using the various feature representations, with the highest and lowest values for each dataset highlighted in green and red respectively.

Video		Feature Representation			
	AE	HOG	LBP	MN2	RGB
Koi	-0.558	-0.315	-0.686	-0.181	-0.378
Pigeons (Square)	-0.456	-0.385	-0.465	-0.321	-0.205
Pigeons (Pavement)	-0.598	-0.444	0.648	-0.257	-0.313
Pigeons (Kerb)	-0.233	-0.178	-0.261	-0.202	-0.200
Pigs	-0.511	-0.426	0.542	-0.219	-0.211

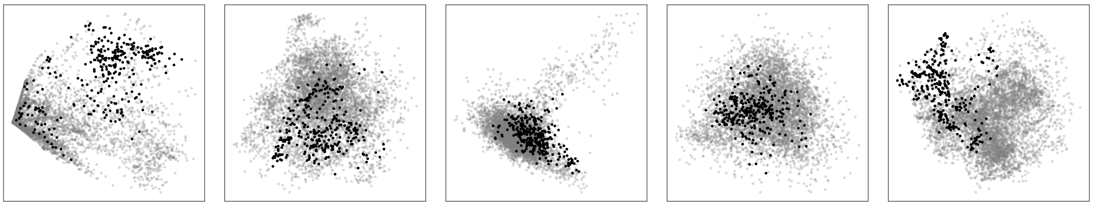
3.3.5 Arbitrarily-shaped clusters

The shape of clusters within a feature space plays a crucial role in determining the effectiveness of a clustering algorithm. As clusters become more irregular or arbitrarily shaped, they often reflect increasingly complex underlying structures in the dataset. Traditional clustering methods, such as k-means, rely on assumptions of convex, similarly shaped clusters—typically spherical—which makes them poorly suited to datasets containing arbitrarily shaped clusters. As a result, applying such methods to complex data can lead to inaccurate or misleading groupings. In contrast, algorithms capable of identifying clusters with varying densities and irregular boundaries are essential in real-world applications, such as image recognition and re-identification, where data structures are often non-linear and intricate. The ability to adapt to such complexity enables algorithms to uncover deeper, more insights from the data.



(a) AE **(b)** HOG **(c)** LBP **(d)** MN2 **(e)** RGB

Figure 3.15: Example of cluster shape from a single consistent identity across all feature representations under PCA dimensionality reduction of the Koi dataset.



(a) AE **(b)** HOG **(c)** LBP **(d)** MN2 **(e)** RGB

Figure 3.16: Example of cluster shape from a single consistent identity across all feature representations under PCA dimensionality reduction of the Pigs dataset.

To illustrate the presence of complex structures within our benchmark dataset, Figure 3.15 and Figure 3.16 depict the spatial distribution of a single identity across all feature representations under PCA dimensionality reduction for the Koi and Pig datasets, respectively. These figures underscore the critical influence of feature representation on clustering outcomes, as they lead to markedly different cluster geometries. Specifically, LBP and MN2 features tend to yield clusters with predominantly convex shapes, whereas AE, HOG, and RGB features give rise to significantly more irregular, non-convex structures. This variation highlights the importance of both feature selection and clustering methodology in the design of effective clustering solutions for complex datasets.

3.4 Experimental Studies

To identify the most effective feature representation for our datasets, we designed an experimental protocol comprising two main stages: feature extraction and classification, as illustrated in Figure 3.17. Feature extraction is performed on the complete set of detections within each video, without consideration of class labels. For classification, a two-fold cross-validation approach is employed to evaluate various state-of-the-art models, where each video is split into two halves. The video is kept intact during splitting to prevent near-identical instances from temporally adjacent frames being divided between training and testing sets. Such splits, which would occur under randomised cross-validation, could lead to artificially inflated accuracy scores.

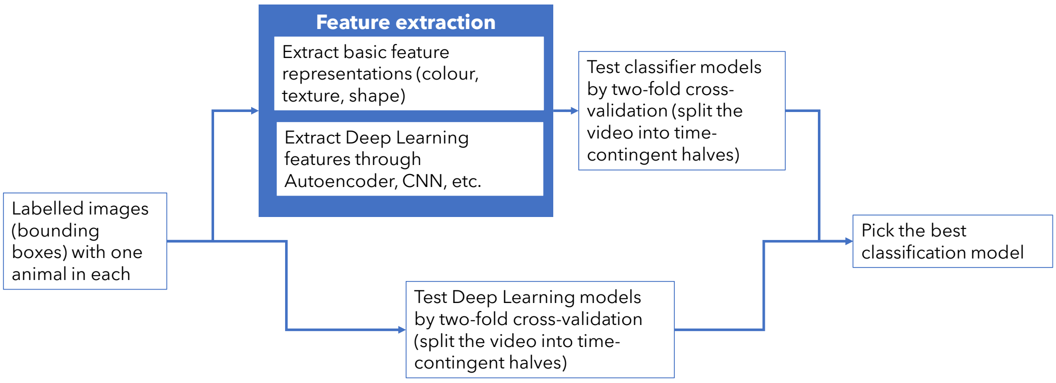

Feature extraction
Extract basic feature representations (colour, texture, shape)
Extract Deep Learning features through Autoencoder, CNN, etc.
Test classifier models by two-fold cross-validation (split the video into time-contingent halves)
Labelled images (bounding boxes) with one animal in each
Pick the best classification model
Test Deep Learning models by two-fold cross-validation (split the video into time-contingent halves)

Figure 3.17: Diagram of the proposed experimental protocol for animal re-identification [114].

We included 23 classifiers from the Python library *lazypredict*, based on *scikit-learn*. These were all the classifiers in this library that could be applied to our data. We grouped the classifiers into: baseline, linear, non-linear, and ensembles, as shown in Table 3.4. Details of these methods can be found in the scikit-learn documentation. These classifiers were applied to the five data representations detailed in Section 3.1. The Largest Prior classifier (Classifier 1 in the Table; also known as Majority or ZeroR classifier) was chosen as a baseline.

Figures 3.18 to 3.22 display glyph plots illustrating classification accuracies, with a separate figure provided for each video. Each figure contains five plots, corresponding to the five different feature representations. Classification accuracy is indicated by the length of the spokes in each plot. The subplots are scaled such that the longest spoke represents the highest accuracy achieved across all feature representations for that particular video, and this maximum is highlighted in red. The average classification accuracy for each feature representation, calculated over all 25 classifiers, is shown in parentheses in the subplot titles. The classifier groups defined in Table 3.4 are distinguished by different shading. Notably, the feature representation with the highest average accuracy does not always yield the best result for a given classifier.

Table 3.4: Classifiers used in this study. The colour boxes correspond to the colours in the figures, with results 3.18 – 3.22. In the electronic version of the document, classifier names include a hyperlink to the classifier implementation documentation.

Baseline

1. Largest Prior classifier (ZeroR/ Majority)

▓ **Linear**

2. Bernoulli (Naïve Bayes)
3. Calibrated CV
4. Gaussian Naïve Bayes
5. Linear Discriminant Analysis
6. Linear SVM
7. Logistic Regression
8. Nearest Centroid
9. Passive Aggressive Classifier
10. Perceptron
11. Ridge Regression
12. Ridge Regression CV
13. SGD

░ **Non-Linear**

14. DecisionTree (C45)
15. Extra Tree
16. K-nn
17. Quadratic Discriminant Analysis
18. SVM

□ **Ensembles**

19. AdaBoost
20. Bagging
21. Extra Tree Ensemble
22. LGBM
23. Random Forest

■ **Deep Learning**

24. Convolutional Neural Network (CNN)
25. Transfer learning using MobileNetV2 (MNV2)

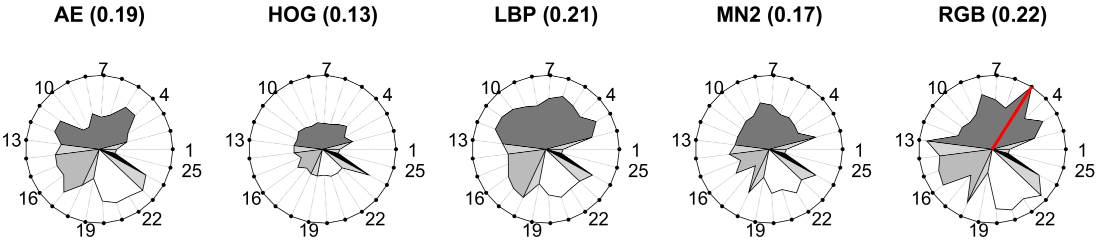


Figure 3.18: Classification accuracy of the 25 classifiers for the five feature representations for the Koi Fish video. Best accuracy of 34.13% was achieved with RGB feature representation and the LDA classifier. [114]

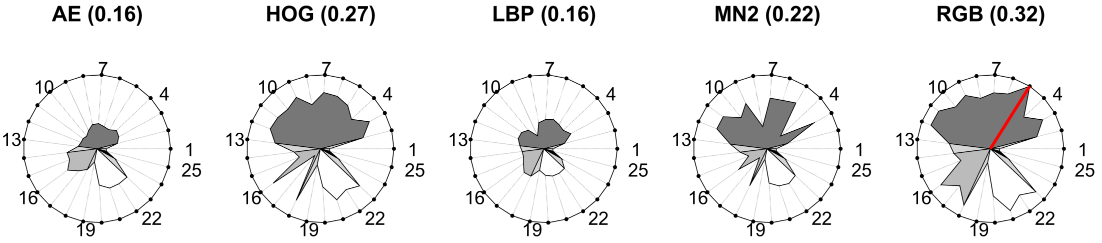


Figure 3.19: Classification accuracy of the 25 classifiers for the five feature representations for the Pigeons (square) video. Best accuracy of 49.13% was achieved with RGB feature representation and the LDA classifier.[114]

Figure 3.23 presents the ranking of feature representations, where each combination of classifier and video is treated as a distinct item. Consequently, each feature representation is associated with 23×5=115 rankings. The figure demonstrates that the RGB representation consistently outperforms the others. Accordingly, RGB will be adopted as the feature representation of choice in all subsequent experimental studies, as well as in the development of clustering solutions aimed at addressing the animal re-identification challenge.

3.5 Summary

This chapter introduced a benchmark dataset developed for the task of animal re-identification from video. To support the development of potential solutions, a comprehensive analysis of the dataset's inherent challenges was undertaken. This included:

- An examination of the number of objects per frame,
- An examination of the degree of occlusion in each video,
- A visual exploration to illustrate concept drift,

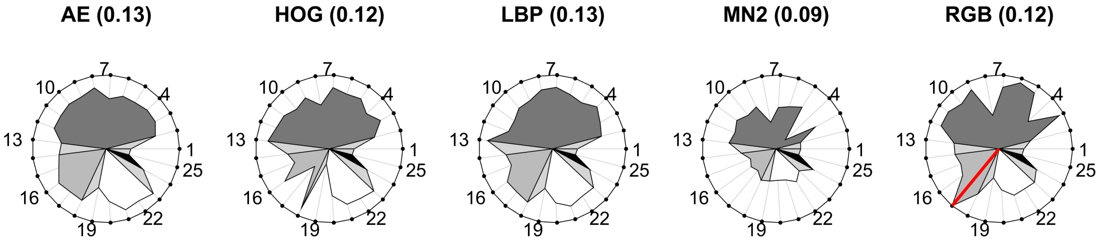


Figure 3.20: Classification accuracy of the 25 classifiers for the five feature representations for the Pigeons (pavement) video. Best accuracy of 18.41% was achieved with RGB feature representation and the QDA classifier.[114]

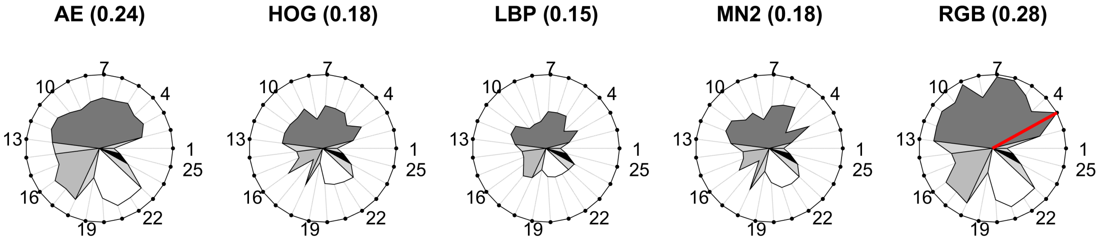


Figure 3.21: Classification accuracy of the 25 classifiers for the five feature representations for the Pigeons (curb) video. Best accuracy of 38.53% was achieved with RGB feature representation and the Calibrated CV classifier.[114]

- An examination of the structural complexity of the data, and the difficulty of separating clusters assessed by internal CVIs,

- A visual illustration showing the presence of arbitrarily shaped clusters.

To complete the analysis of the data collection, an experimental study was conducted. Its main purpose was to determine the most effective feature representation for the dataset. As a by-product, the classification experiment gives us an idea about the achievable classification accuracy in such a type of data. The results indicated that colour-based RGB features consistently outperformed alternative representations. Consequently, RGB features are recommended as the preferred representation in all subsequent experimental investigations and in the development of clustering methodologies for addressing the animal re-identification task.

Table 3.5 summarises the key findings presented in this chapter. It reports the highest classification accuracy achieved for each dataset, all of which were obtained using the RGB feature representation. The table also includes the overall occlusion within each video, expressed as the percentage of an object that is occluded, the silhouette score for each dataset under the RGB feature representation—reflecting the structural complexity of the data—and the average number of objects per frame across the videos.

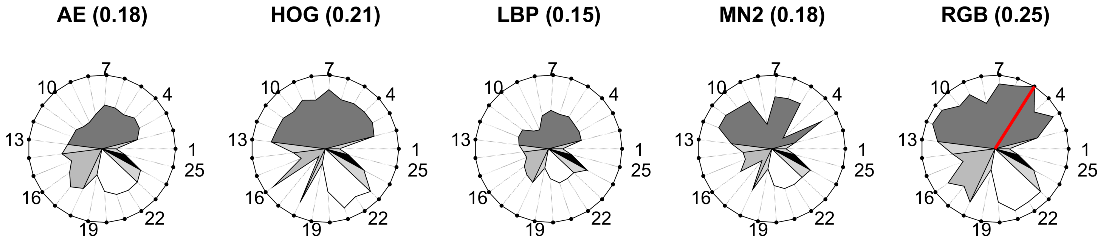


Figure 3.22: Classification accuracy of the 25 classifiers for the five feature representations for the Pigs video. Best accuracy of 34.51% was achieved with RGB feature representation and the LDA classifier.[114]

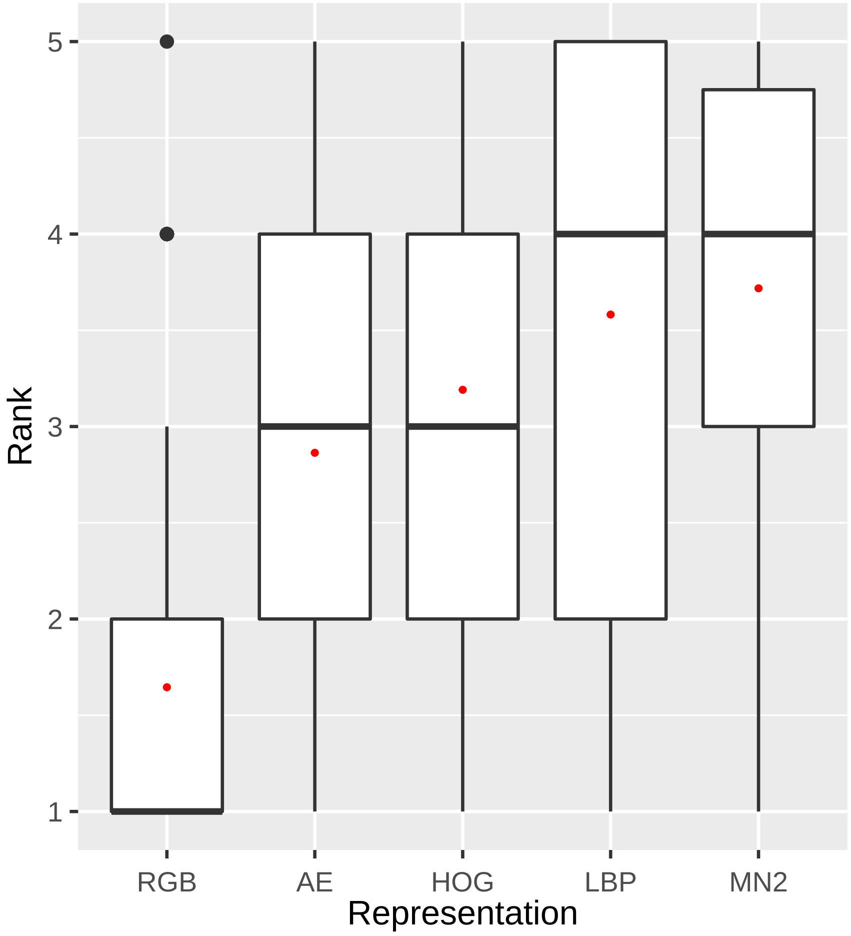


Figure 3.23: Box plot illustrating the comparative ranking of the five feature representations, arranged from best (left) to worst (right) based on their mean performance, indicated by the red dot. The plot conveys the distribution of rankings across multiple evaluations, highlighting variability and relative consistency among the feature representations.

[114]

From Table 3.5, the relationship between the dataset challenges and classification accuracy using the RGB feature representation becomes evident. A general trend can be observed whereby increased levels of occlusion correspond to a reduction in classification accuracy. However, the Koi and Pigeons (Kerb) datasets present exceptions to this trend, exhibiting an inverse relationship. This discrepancy can be explained by the structural complexity of the datasets, as indicated by the silhouette scores. Although

Table 3.5: Summary of the key findings throughout the chapter

	Koi		
Classification Accuracy	Overall Occlusion (%)	Silhouette Score	Average Objects p/f
34.13%	5.09	-0.378	3.1

	Pigeon (Square)		
Classification Accuracy	Overall Occlusion (%)	Silhouette Score	Average Objects p/f
49.13%	2.68	-0.205	16.3

	Pigeon (Square)		
Classification Accuracy	Overall Occlusion (%)	Silhouette Score	Average Objects p/f
18.41%	8.48	-0.313	5.1

	Pigeon (Kerb)		
Classification Accuracy	Overall Occlusion (%)	Silhouette Score	Average Objects p/f
38.53%	7.93	-0.200	10.6

	Pigs		
Classification Accuracy	Overall Occlusion (%)	Silhouette Score	Average Objects p/f
34.51%	6.09	-0.211	12.4

the Koi dataset exhibits lower levels of occlusion compared to the Pigeons (Kerb) dataset, its greater structural complexity results in a lower classification accuracy. This suggests that structural complexity may, in certain cases, have a more pronounced impact on performance than occlusion alone.

Chapter 4

Bounding Box Detection

The development of a fully autonomous pipeline for animal re-identification necessitates the integration of accurate and dependable components in order to be viable for real-world deployment. The most critical element in both online and offline pipelines is the capacity to detect animals within video footage. In the absence of a reliable object detection mechanism, the credibility of the pipeline's output is significantly undermined. It is therefore imperative to employ the highest-performing object detector available, as its accuracy has a direct impact on all subsequent stages. An unreliable detector may generate false positives, fail to identify key instances, or produce incomplete BBs—leading to feature representations of irrelevant or partial objects. Such flawed inputs can introduce outliers into the clustering process, thereby impairing the overall performance of any method utilised for animal re-identification.

This chapter introduces a novel object detection approach that integrates both detection and tracking paradigms, with the aim of harnessing the strengths of each while addressing their respective limitations. This integrated method enhances the pipeline by providing a more accurate object detection framework, demonstrating improved performance as measured by the Average Precision (AP) metric, which is commonly used to evaluate the efficacy of object detection algorithms.

Contributions covered in this Chapter

Combining bounding boxes output from an object detector and an object tracker as a from of ensemble to improve the overall detection of animals in video frames

Publication 4 from Section 1.4

2

4.1 Methodology

While the field of object tracking and detection is continually advancing and has likely progressed since this study was undertaken, the most up-to-date benchmark methods available at the time were employed. These included approaches for both tracking and detection, obtained from "Papers with Code", a reputable platform offering machine learning resources such as academic publications, source code, and datasets. Specifically, the detection method utilised was the MMDetection (MMDet) object detector, and the tracking method implemented was UniTrack.

4.1.1 Object Detection

MMDetection [50] (MMDet) is an object detection toolbox that includes a comprehensive collection of object detection and instance segmentation methods, along with associated components and modules. The code and modules are available at [1]. Although the model architectures of various detectors differ, they share common components, which can be broadly categorised into the following classes:

- **Backbone**. The Backbone is the component of the model responsible for transforming an image into its corresponding feature map.

- **Neck**. The Neck is the component that links the backbone to the heads. Its purpose is to refine or reconfigure the raw feature maps produced by the backbone.

- **DenseHead**. The DenseHead is the component that operates on densely distributed regions of the refined or reconfigured feature maps.

- **RoIHead**. The RoIhead is the component that extracts region-of-interest (RoI) features from one or more feature maps using RoI pooling-like operations.

The MMDetection toolbox provides a wide range of object detectors, spanning from single-stage to two-stage architectures, along with an extensive selection of components for each part of the detection pipeline. Although there were numerous options available, we chose to adopt a two-stage object detection architecture due to its superior precision

[1] `https://github.com/open-mmlab/mmdetection`

and more accurate localisation, particularly in complex or cluttered scenes. Figure 4.1 shows an illustration of the steps involved in two-stage bounding box detection.

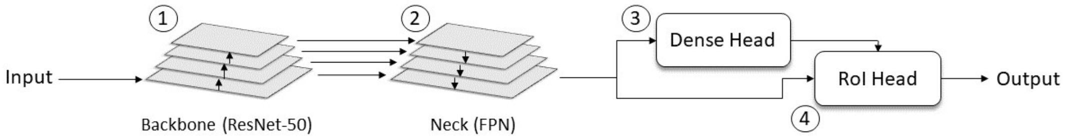


Figure 4.1: Illustrates the two-stage detection pipeline of the MMDet toolbox. The input image is first processed by the backbone to generate a raw feature map, which is subsequently refined by the neck. The refined feature map is then passed to both the dense head and the RoI head, which focus on dense regions and regions of interest, respectively. The RoI head additionally utilises the output of the dense head and ultimately produces the bounding boxes of objects present in the image.

4.1.2 Tracking

UniTrack [179] is among the top-performing methods on benchmark datasets commonly used by the MOT community, offering a unified solution that addresses five distinct tasks within a single framework. It features a single, task-agnostic appearance model that can be trained either in a supervised or self-supervised manner, alongside several ’head’ components designed for specific tasks, which do not require additional training. This versatile framework supports a range of applications, including Single Object Tracking (SOT), Video Object Segmentation (VOS), and MOT. For our experiments, we focus on the MOT component of the framework.

The UniTrack framework approaches all tracking tasks as a combination of two fundamental components. The first is a propagation component, which estimates the object’s state—such as its bounding box, mask, or pose—in the current frame based on information from the previous frame. The second is an association component, which matches and identifies objects across frames using various appearance features.

The MOT tracking process in UniTrack comprises two primary stages. The first involves the Appearance Model, which converts the 2D video frame into a feature map; in our implementation, we used the recommended ’default’ YOLOX detector. The second stage is Association, where features from the current frame are matched with those in adjacent frames to construct object tracks. This is done using a reconstruction-based similarity metric, which evaluates how effectively the features of one object can be reconstructed from another—an approach that improves resilience to occlusion, pose changes, and noise. A distance matrix is generated between existing tracks and new detections, and the Hungarian algorithm is applied to establish the optimal matches across frames. For this stage, we also adopted the ’default’ setting, utilising Imagenet-ResNet18-s3. The MMDetector produces BBs for each frame, each assigned a unique track ID.

4.1.3 Proposed Fusion Method

Our preliminary experiment showed that the detector returns duplicate BBs. Occasionally, it also returns inadequately small BBs. Therefore, we set up a percentile threshold P and removed the smallest false positive BBs from the detector output.

The next phase is aggregating the BBs from the two outputs. To complete this phase, we apply the following steps for each frame F_t:

1. Identify the BBs in frame F_t returned by the detector. Denote this list by $\mathcal{B}^{det}$. Identify the BBs in frame F_t returned by the tracker. Denote this list by $\mathcal{B}^{tr}$. Pool together the two lists into a single list $\mathcal{B} = \mathcal{B}^{det} \cup \mathcal{B}^{tr}$.

2. Calculate a square matrix $\mathbf{M}_{|\mathcal{B}|,|\mathcal{B}|}$ with IoU values between all pairs of BBs in $\mathcal{B}$. Set the main diagonal of $\mathbf{M}$ to zeros to eliminate the match between each box with itself.

3. Apply a duplicate threshold D on the values of M. All pairs of BBs whose IoU is greater than D are perceived to be the same bounding box. This transforms $\mathbf{M}$ into a binary matrix $\mathbf{M}^{bin}_{|\mathcal{B}|,|\mathcal{B}|}$.

4. Considering $\mathbf{M}^{bin}$ as an adjacency matrix of a graph, identify the connected components. Each component is fused into a single bounding box. The fusion takes the minimum top left corner (on both coordinates) and the maximum bottom right corner (on both coordinates). The detector output contains a value of certainty attached to each bounding box, while the tracker output places the same certainty to all boxes. To calculate the certainty of a fused bounding box (connected component), we take the maximum certainty of the boxes being fused.

The parameters of our combination methods are the percentile threshold P and the duplicate threshold D. Below we carry out grid-like experiments to demonstrate that the proposed method is capable of outperforming both the detector and the tracker taken individually.

4.2 Experimental Study

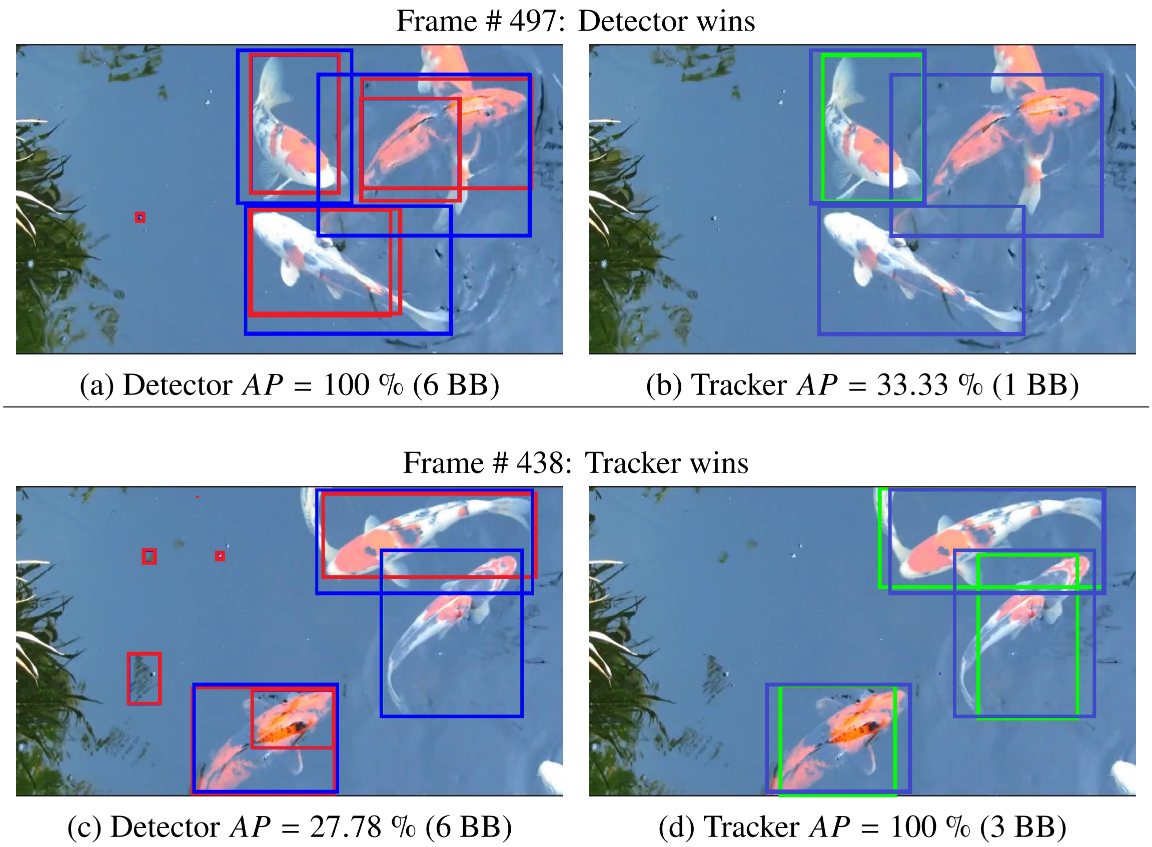
Our experimental study consists of two parts. First, we evaluate the performance of the object detector (MMDet) and the MOT tracker (UniTrack) using their default settings and configurations to ensure a fair comparison between the two. The second part involves comparing our proposed fusion method against both the detector and the tracking approaches. Each stage of the experiment was carried out using the benchmark datasets discussed in Chapter 3.

4.2.1 Comparison between the Detector and the Tracker

To demonstrate the motivation behind our experiment and provide insight into our fusion approach, we highlight the discrepancies between the object detector and the MOT tracker in Figure 4.2. We compute the AP for both methods on a frame-by-frame basis. Plots (a) and (b) present the same video frame with outputs from the object detector (blue boxes) and the MOT tracker (red boxes), compared against the ground truth (green boxes). In this case, the detector outperforms the tracker. In contrast, plots (c) and (d) depict a different frame from the video, again showing outputs from both methods alongside the ground truth. Here, the MOT tracker demonstrates superior performance compared to the detector.

The frame-by-frame variability in performance, as illustrated in Figure 4.2, highlights the need for a more consistent approach to object detection. Since neither method consistently outperforms the other, the most effective solution—when both outputs are available—is to adopt a fusion strategy that leverages the strengths of each while compensating for their individual shortcomings.

To further highlight the differences between the two approaches, we analysed the number of BBs generated by each method per frame across the five videos in our benchmark dataset. Table 4.1 presents the minimum, maximum, and average number of BBs



(a) Detector AP = 100 % (6 BB) (b) Tracker AP = 33.33 % (1 BB)

(c) Detector AP = 27.78 % (6 BB) (d) Tracker AP = 100 % (3 BB)

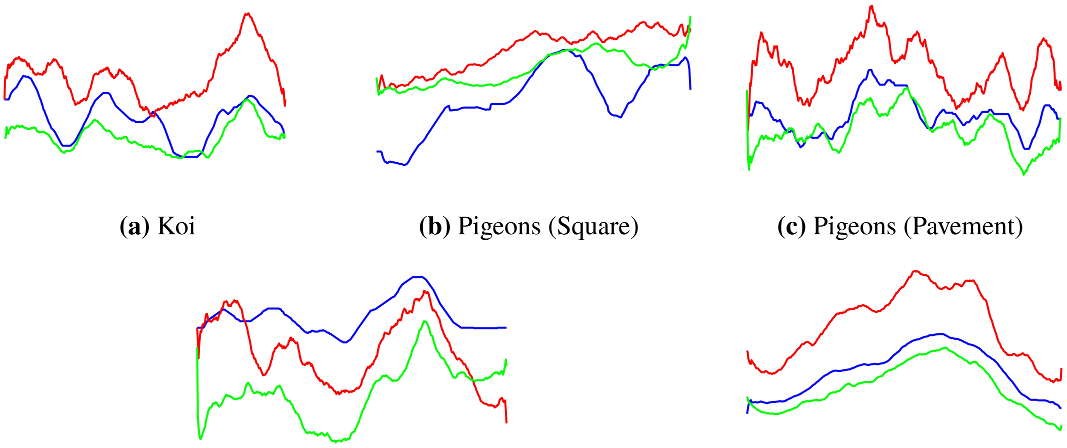
Figure 4.2: Example from the Koi fish video of differences in object detection between the Detector and the Tracker methods. The ground truth is shown with blue, the Detector results, in red, and the Tracker results, in green. In both images there are three ground truth BBs. [187]

produced by both the detector and the MOT tracker, with ground truth values included for comparison. The table reveals that the detector frequently overestimates the number of objects, while the MOT tracker tends to underestimate them.

Table 4.1: Number of detections from MMDet (MM) and UniTrack (UT) for each video. The ground truth (GT) value is given for comparison.

Video	Min/frame			Max/frame			Avg/frame		
	GT	MM	UT	GT	MM	UT	GT	MM	UT
Koi Fish	1	1	0	6	11	6	3.1	5.3	2.1
Pigeons (Ground)	3	2	1	8	15	7	5.1	6.8	4.6
Pigeons (Kerb)	8	3	1	13	16	11	10.6	8.8	6.6
Pigeons (Square)	9	14	13	23	28	24	16.3	20.2	18.6
Pigs	4	8	2	20	37	18	12.4	20.4	9.6

To further explore the number of BBs produced by both methods, Figure 5.5 plots the ground truth counts (blue), detector counts (red), and MOT tracker counts (green) against the frame number. Each curve has been smoothed using a window size of 40 frames. The plots show that the detector consistently produces more BBs than the MOT tracker. However, the tracker's output is generally closer to the ground truth, with the exception of the Pigeons (Kerb) video, where the detector more accurately aligns with the ground truth values.



(a) Koi

(b) Pigeons (Square)

(c) Pigeons (Pavement)

(d) Pigeons (Kerb)

(e) Pigs

Figure 4.3: The x-axis is the frame number and the y-axis is the number of BBs per frame. The blue curve is the ground truth, the red is the detector output, and the green is the tracker output. [187]

As illustrated in Figure 4.2, the detector can produce noisy or duplicate BBs, which may account for the higher detection counts per frame. However, this increase does not substantially affect the AP, as the metric generally disregards near-duplicate or excessively small BBs.

To better assess overall performance, AP was calculated for both methods across all five videos, with the results displayed in Figure 4.4. Interestingly, although the tracker generally produced a bounding box count more closely aligned with the ground truth, the AP metric consistently favoured the detector—except in the Koi Fish video. This highlights that performance depends not only on detecting the correct number of objects, as achieved by the MOT tracker, but also on the precision of object localisation within each bounding box, where the detector appears to excel. These findings further emphasise the disparity between the two approaches and reinforce the motivation for developing a fusion method that leverages the strengths of both.

4.2.2 The Fusion Method

The proposed fusion method was applied to the outputs of both the detector and the MOT tracker for each video in our benchmark dataset. To assess the impact of the

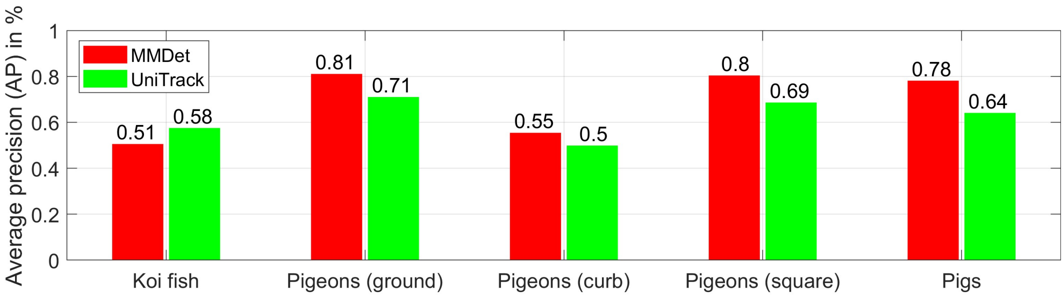


Figure 4.4: AP scores for object detection are presented for the five videos, with MMDetection (MMDet) indicated in red and UniTrack indicated in green. The plot enables a direct comparison of the performance of the two methods across different video sequences, highlighting variations in detection accuracy and consistency. [187]

two key parameters—the percentile threshold P and the duplicate threshold D—the experiment was run on each dataset using every unique combination of parameter values, with $D = 0.40, 0.45, 0.50, \ldots, 0.95$ and $P = \{1, 3, 5, 10, 15, 20, 25, 30\}$. The results of these experiments report the AP for each datasets and are visualised in Figures 4.5 and 4.6.

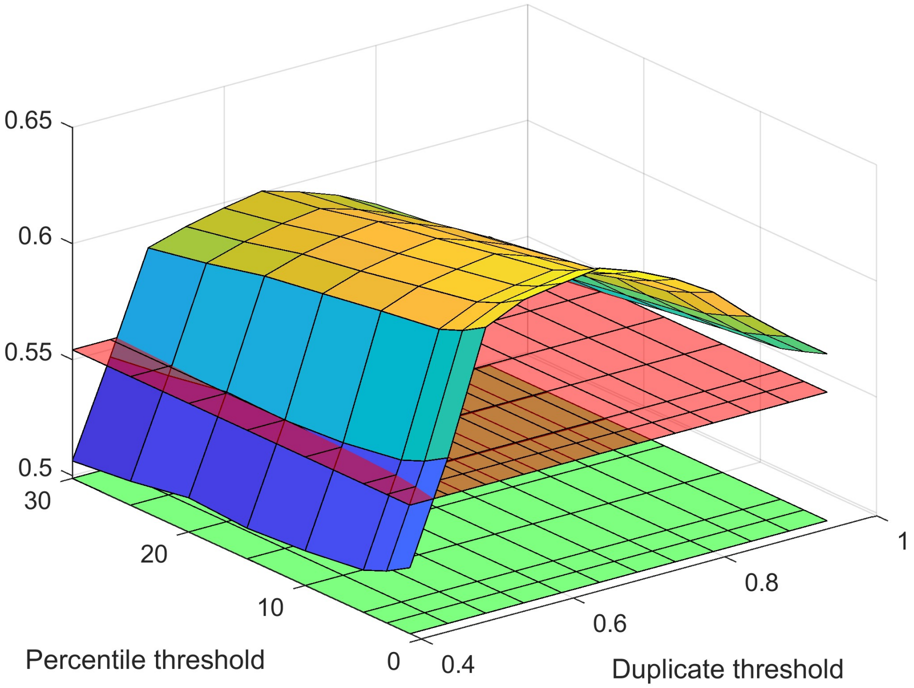
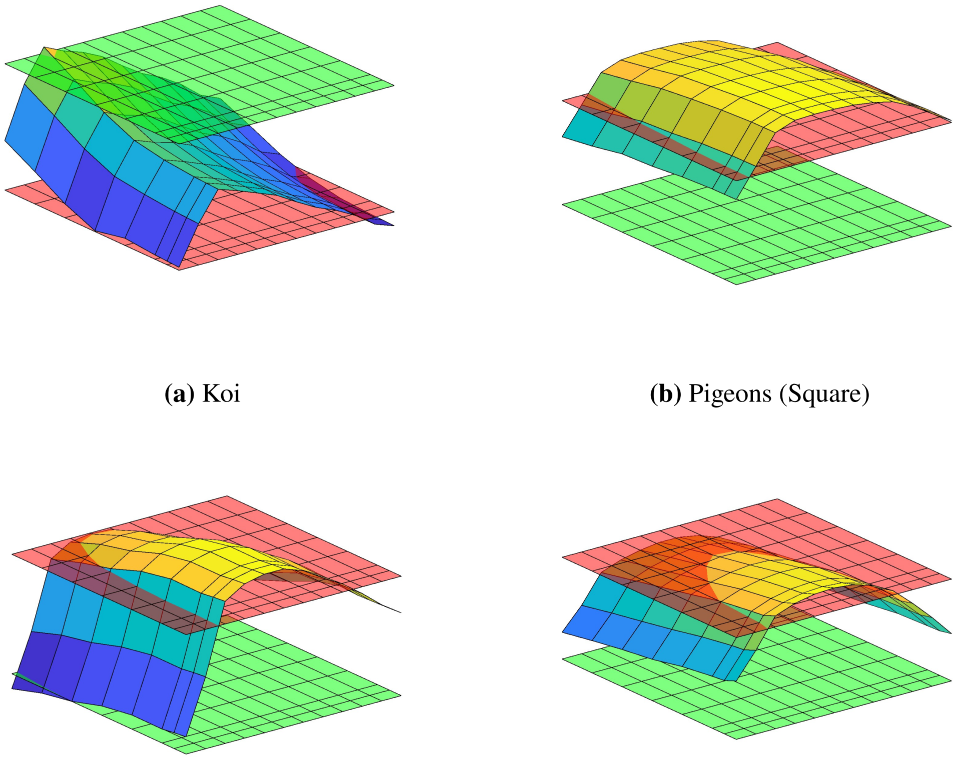


Figure 4.5: Example of the AP obtained by the combination method in comparison with the AP of the detector (red plane) and the tracker (green plane) for the Pigeons (Kerb) video. The surface is drawn in the space of values spanned by the duplicate threshold D and the percentile threshold P.[187]

Figure 4.5 illustrates that the AP values achieved using the fusion method exceed those of both the detector and the tracker across the majority of parameter combinations.



(a) Koi

(b) Pigeons (Square)

(c) Pigeons (Pavement)

(d) Pigs

Figure 4.6: AP surface obtained from the combination method for different parameter values P and D. The red plane is the constant AP value for the detector and the green plane is the constant AP value for the tracker.[187]

Figure 4.6 presents the corresponding performance surfaces for the remaining four videos. While the improvements over the individual methods are less pronounced than in the example shown in Figure 4.5, noticeable gains are still evident in the Pigeons and Pigs videos. In the Koi Fish video, the fusion method slightly outperforms the tracker, reaching a peak AP of 0.5736 only at $P = 30$ and $D = 0.50$.

Figure 4.6a demonstrates a markedly different behaviour of the proposed fusion method, a change that can be attributed to the elevated false positive rate of the detector, as shown in Figure 4.3a. The topology of the AP surface suggests that the false positive percentile threshold P influences the overall performance of the method only in cases where false positives are highly probable. In all other scenarios, the threshold P appears either to be irrelevant or to diminish performance.

4.3 Summary

This chapter examined the strengths and weaknesses of object detection and object tracking paradigms, and introduced a fusion approach that integrates the outputs of both methods. By combining the results from the object detector and the tracker, it was possible to leverage the advantages of each while mitigating their respective limitations. The proposed fusion method was shown to outperform the individual components across all videos in our benchmark animal dataset, as outlined in Chapter 3.

Chapter 5

Online Animal Re-Identification

An online solution for animal re-identification offers a valuable tool for agriculture professionals and ecologists, enabling real-time tracking and monitoring of individual animals without the need for physical tagging or manual oversight. By facilitating continuous, non-intrusive observation through live video analysis, such systems support more efficient livestock management, enhance wildlife conservation efforts, and improve the quality of ecological research. Real-time data allows for the prompt detection of behavioural changes, health issues, and movement anomalies, enabling timely interventions and promoting better animal welfare outcomes. This approach not only increases operational efficiency but also supports more ethical and sustainable practices across agricultural and ecological domains. Furthermore, the automated analysis of live video footage enables scalable monitoring in complex, real-world environments, allowing for informed decision-making and effective resource allocation in animal health and behavioural studies.

Developing a solution for real-time re-identification presents distinct challenges when compared to offline settings. Any approach designed for real-time application must give careful consideration to data summarisation, as the potentially unbounded nature of data streams makes complete storage impractical. To ensure the effectiveness of such a solution, it is crucial to minimise the volume of stored data without compromising the performance of the clustering algorithm. In addition to the inherent difficulties of online processing, the various challenges associated with the datasets—outlined in Section 3.3—must also be taken into account when devising a solution for real-time analysis.

This chapter explores and compares the effectiveness of hierarchical and centroid-based constrained clustering for animal re-identification, highlighting the performance disparities observed when applied to varying data window sizes. Furthermore, it proposes a novel real-time constrained clustering solution to advance the field of animal re-identification from live video footage as a part of the proposed fully autonomous re-identification system.

Contributions covered in this Chapter

Comparing hierarchical and non-hierarchical clustering for complex data configurations present in animal data & Proposing an online constrained clustering solution for species-invariants animal re-identification.

Publications 5 & 8 from Section 1.4.

4

5.1 Hierarchical clustering Vs Centroid-based clustering

This section presents a comparative analysis of constraint-based hierarchical and centroid-based clustering applied to the benchmark dataset. As video data frequently gives rise to arbitrarily shaped clusters it is crucial to determine which clustering approach is best suited to accommodate such complexity in both online and offline contexts.

By contrasting hierarchical and centroid-based techniques, we aim to determine the most suitable method for the animal re-identification task. Hierarchical clustering tends to perform better with elongated, string-like clusters, while centroid-based methods are typically more effective with convex-shaped clusters. While both types of structures appear within the RGB feature space, cluster shape alone does not provide a conclusive basis for method selection. This study seeks to empirically assess performance, shedding light on the dominant structural patterns present in the data.

Importantly, the use of video footage allows us to derive instance-level constraints in the form of must-link and cannot-link relationships. These constraints can be seamlessly integrated into both clustering approaches to enhance their accuracy. Since such constraints can be automatically generated from video data—without the need for domain expert annotation—it would be a missed opportunity to overlook this valuable source of information when designing or evaluating clustering algorithms for re-identification from video.

5.1.1 Preliminaries

This study adopts an online processing approach, whereby each video in the benchmark dataset is divided into consecutive windows of equal size, ranging from 2 to 20 frames. Each window thus represents a small portion of the dataset, containing the objects present across a limited temporal span—from as few as two consecutive frames to as many as twenty. These windows are processed independently using a set of comparative methods, with any remaining frames that do not constitute a complete window excluded from analysis. For each window size, all methods are applied across all segments, yielding a corresponding metric value for each window. As the number of frames per window decreases, the structural complexity of the data is reduced. Applying hierarchical and centroid-based clustering algorithms to these simplified segments facilitates the identification of prominent structures that may otherwise be obscured by the complexity of the full dataset.

It should be noted that, although this online processing approach reduces the structural complexity of the dataset, smaller window sizes may not contain sufficient data to enable the formation of informative clusters. Conversely, larger windows may encompass more complex or overlapping clusters, which can be difficult to distinguish, as indicated by the dataset analysis in Chapter 3.

5.1.2 Constraint Generation

Constraints are traditionally derived through manual annotation by domain experts—a process that is both time-consuming and prone to inaccuracies and human error. However, when using video data, such constraints can instead be inferred automatically from the spatial and temporal properties of object detections. By leveraging the temporal and spatial information of each detection, we can determine with high confidence whether two objects should be must-linked or, conversely, cannot-linked.

CL constraints

CL constraints are straightforward to generate, as they rely solely on temporal information and can be derived on a frame-by-frame basis. For a given video frame F_t, which contains one detection per object—represented as $\mathcal{D}_t = \{D_{t,1}, D_{t,2}, \ldots, D_{t,M_t}\}$, where M_t detections have been identified—it can be inferred that these detections occurred simultaneously. Assuming the detector provides a single, accurate detection per object, it follows that each detection corresponds to a distinct individual, thereby allowing for the construction of $\frac{M_t(M_t-1)}{2}$ CL constraints with a high degree of confidence.

To illustrate the procedure for generating CL constraints, consider the example depicted in Figure 5.1. The frame F_t comprises five unique detections (using the ground truth labelling), each delineated by a bounding box surrounding a different fish. Under the assumption that each detection pertains to a separate individual, CL constraints can be established between every possible pair of detections within the frame.

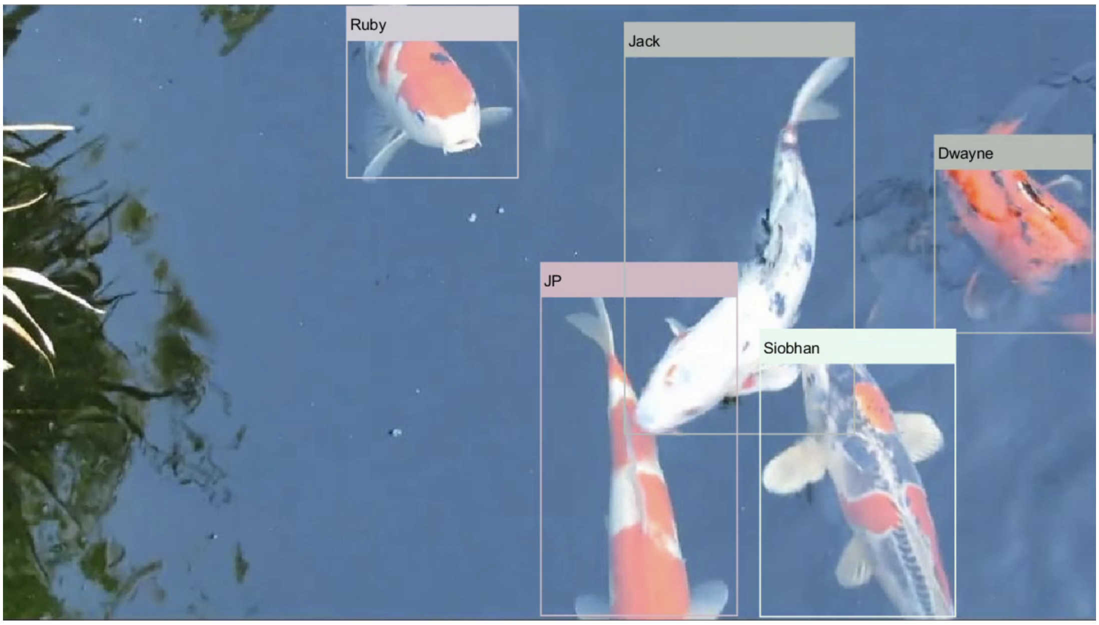


Figure 5.1: Example of an annotated frame from a Koi video containing five distinct identities, each labelled with its identity and enclosed within a bounding box.

This relationship is depicted in Figure 5.2 as a complete graph $G = (V, E)$, where each node $v \in V$ represents a detection, and each edge $e \in E$ denotes a CL constraint between a pair of detections. The set of CL constraints denoted by $C\mathcal{L}$ are derived from the frame, and thus corresponds directly to the edge set E of the graph G.

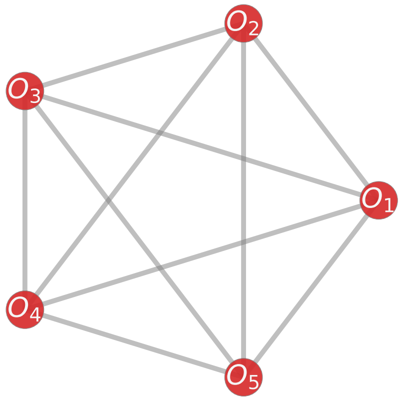


Figure 5.2: Illustration of the CL relationship within a frame, where detections are represented as nodes and CL constraints are represented as edges.

ML constraints

To generate the set of ML constraints denoted by $\mathcal{ML}$, more information is required. They rely not only on temporal relationships but also on spatial relationships between detections. By examining pairs of consecutive frames (F_t, F_{t+1}) and the BBs detected in each-denoted by $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$—we can evaluate pairs of detections that do not appear in the same frame. For each such pair, we compute the intersection over union (IoU) to assess their spatial similarity. Since the frames are consecutive, the object's location is unlikely to have shifted significantly between detections, and the degree of overlap between BBs can help indicate whether the detections correspond to the same object.

To illustrate the process of creating ML constraints, let F_t and F_{t+1} denote the consecutive frames shown in Figure 5.3, each containing three detections. In this example, the detections have been annotated with their identities; however, the calculations only require the BBs. The identities are included solely to aid comparison.

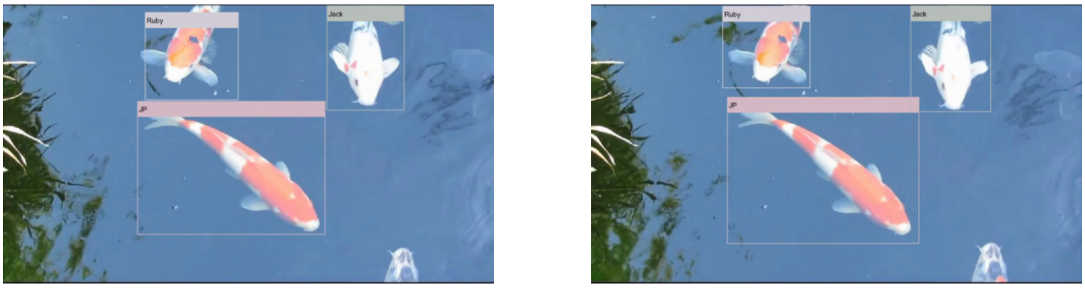


Figure 5.3: Example of two consecutive frames from the Koi dataset, illustrating the temporal continuity of object appearances and the spatial positions of the annotated identities across successive frames.

Figure 5.4 presents $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$ together in a single plot, aligned with the reference frame, to visualise the overlap between BBs across both frames. Let the detections in frame F_t (on the left) be represented by $\mathcal{D}_t = \{A, B, C\}$, shown in red, and those in frame F_{t+1} (on the right) by $\mathcal{D}_{t+1} = \{D, E, F\}$, shown in green.

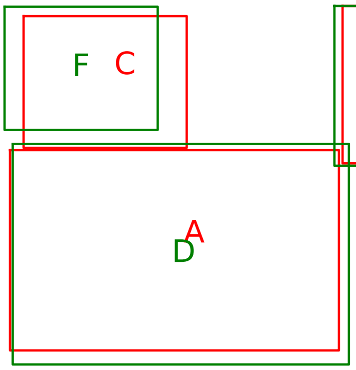


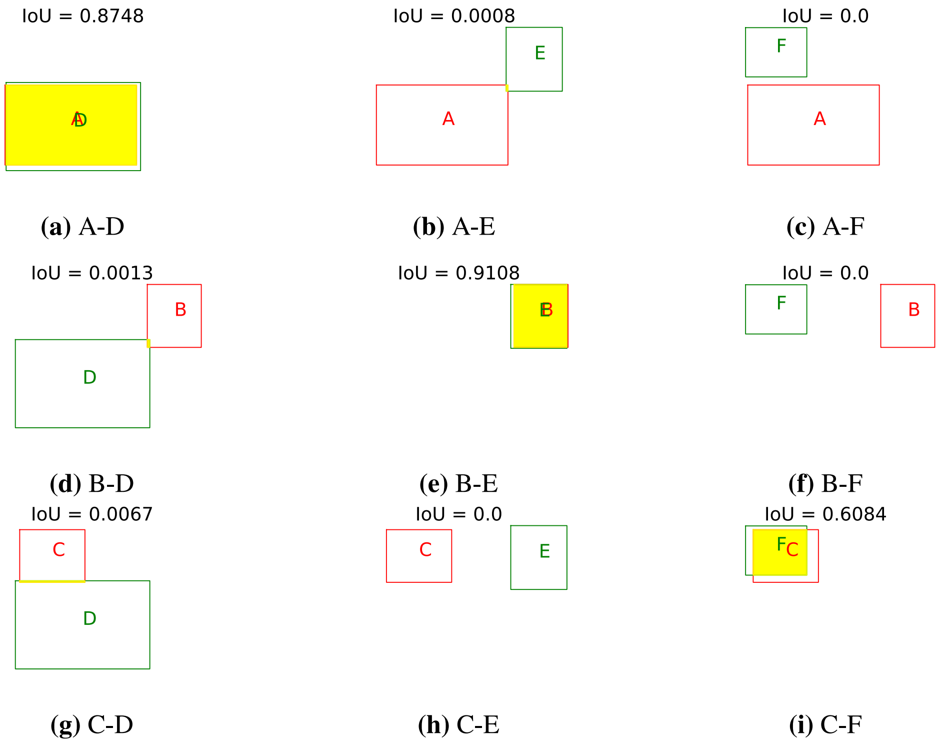
Figure 5.4: Bounding boxes from two consecutive frames, where red boxes denote the first frame and green boxes represent the subsequent frame. Each bounding box is positioned with reference to its respective frame and is labelled with a letter for identification.

A comparison is then performed between each bounding box in $\mathcal{D}_t$ and each in $\mathcal{D}_{t+1}$, where the area of intersection for each pair is calculated and divided by the area of their union to compute the Intersection over Union (IoU). Figure 5.5 illustrates these comparisons, with the overlapping region highlighted in yellow. Each subplot also displays the corresponding IoU value at the top of the plot.

To aid the visualisation we can represent the comparisons between BBs as a weighted graph $G = (V, E, W)$, as shown in Figure 5.6. In this graph, each node in $v \in V$ corresponds to a bounding box from a frame, with node colours consistent with those in Figure 5.5. Each edge $e \in E$ represents a comparison between BBs, and the weight $w \in W$ assigned to an edge denotes the IoU value between the corresponding BBs.

Once all comparisons have been completed and an IoU value has been calculated for each, a threshold τ_{ML} can be applied to the edges $e \in E$. Any edge with a weight less than τ_{ML} is removed from the graph. The resulting graph is shown in Figure 5.7, and it highlights the ML constraints between the consecutive frames F_t and F_{t+1}.

In this example, there are no competing edges—each detection in F_t is connected to only one detection in F_{t+1}. However, in cases where multiple edges exceed the threshold



(a) A-D **(b)** A-E **(c)** A-F

(d) B-D **(e)** B-E **(f)** B-F

(g) C-D **(h)** C-E **(i)** C-F

Figure 5.5: Illustrates the IoU calculations for each pair of labelled bounding boxes across consecutive frames, with the intersecting region highlighted in yellow. The corresponding IoU value is displayed at the top of each subplot.

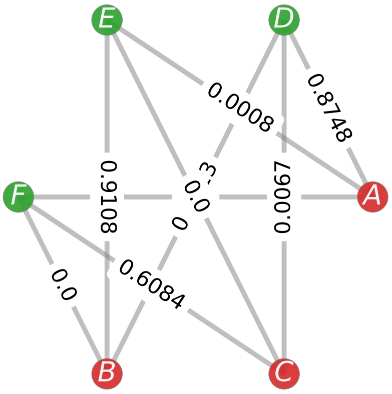


Figure 5.6: Illustrates a weighted graph of detections and their IoU values. Each node represents a bounding box detection with its associated label and is coloured according to the frame in which it appears. The edges are weighted by the corresponding IoU value.

τ_{ML}, the edge with the highest weight (i.e., the highest IoU) is retained to ensure only a single ML constraint exists between a single pair of detections.

Although the displacement of detections across successive frames may fluctuate due to factors such as camera motion, frame rate, or the velocity of the tracked subject, these variables are typically controlled in standard animal tracking contexts. Camera movement is generally minimised, frame rates remain consistent, and, provided the

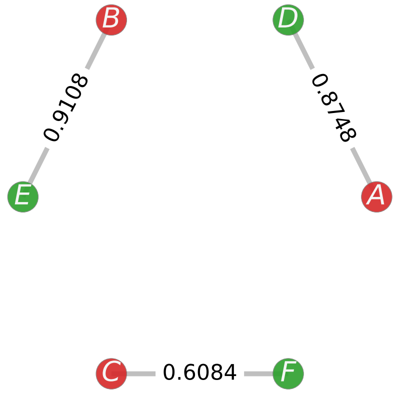


Figure 5.7: Weighted graph of detections and their IoU values after the application of thresholding, with non-conforming edges removed. Each node corresponds to a bounding box detection with its associated label and is coloured according to the frame in which it appears. The remaining edges are weighted by their respective IoU values.

animals are not excessively fast-moving, their inter-frame displacement tends to be relatively limited. Each of these potential sources of variation may impact the choice of the overlap threshold, τ_{ML}, required to reliably infer that two detections correspond to the same object.

In cases where the animals move at such speed that the displacement of bounding boxes between consecutive frames yields an IoU value of zero, a more advanced approach is necessary to derive the ML constraints. A viable strategy would be to estimate the animal's trajectory and interpolate bounding boxes along this path, thereby enabling the calculation of IoU values through comparison between the interpolated bounding boxes and the trajectory's end points.

5.1.3 Experimental Study

This section presents the experimental study conducted to evaluate and compare the effectiveness of hierarchical and centroid-based constrained clustering methods on the benchmark dataset outlined in Chapter 3.

Methods Compared

- Constrained Agglomerative Hierarchical clustering: Klein *et al.* [102] argued that the incorporation of constraints into hierarchical clustering methods does not necessitate alterations to the unsupervised algorithm itself. Owing to the intrinsic properties of hierarchical techniques—namely, the ability to derive a clustering partition from a distance matrix—it is feasible to integrate constraints by modifying the distance matrix accordingly.

Firstly, the set of ML constraints ($\mathcal{ML}$) does not necessarily represent a complete collection. There may exist object pairs that, while not explicitly included in $\mathcal{ML}$, are implicitly linked through transitive closure. Consequently, we expand $\mathcal{ML}$ to $\mathcal{ML}a$ so as to incorporate all object pairs that should be connected via ML relations. To achieve this, a graph is constructed with N nodes, where N is the number of data points. An edge is placed between each pair of nodes specified by $\mathcal{ML}$. The connected components of this graph are then identified. For instance, if the pairs (i, j) and (j, k) are present in $\mathcal{ML}$, the associated connected component will comprise all three objects: x_i, x_j, and x_k. Accordingly, $\mathcal{ML}$ is augmented with all pairwise combinations within each connected component.

Let $\mathbf{M}_{N,N} = (M_{ij})$ denote a distance matrix, where M_{ij} represents the distance between data points x_i and x_j. Constraints are embedded into $\mathbf{M}$ by adjusting the entries M_{ij} in accordance with the constraints applicable to points x_i and x_j. Specifically, if $(i, j) \in \mathcal{ML}a$, then $M_{ij} = 0$; if $(i, j) \in \mathcal{CL}$, then $M_{ij} = \infty$.

Following the modification of the distance matrix to reflect the sets $\mathcal{ML}a$ and $\mathcal{CL}$, conventional unsupervised hierarchical clustering algorithms—such as average linkage, single linkage, and complete linkage—can be applied to generate a partitioning of the data. The respective constrained variants of these methods will be referred to as CAL (Constrained Average Linkage), CCL (Constrained Complete Linkage), and CSL (Constrained Single Linkage).

- COP K-Means: Wagstaff *et al.* [173] introduced COP K-Means, a variant of the widely used centroid-based clustering method K-Means, designed to accommodate pairwise constraints such as ML and CL. The algorithm is presented in Table 5.1. Since the order of points to be assigned to clusters is random, COP k-means often ends prematurely, without returning a viable solution.

- Pairwise Confidence Constraints Clustering (PCCC): Beumann *et al.* [20] presented the PCCC algorithm, offering users the flexibility to specify pairwise constraints as either hard constraints (ML, CL), which must be strictly adhered to,

Table 5.1: COP K-means Algorithm

```
Algorithm: COP-kmeans(Dataset $\mathcal{X}$, must-link constraints $\mathcal{ML}$, cannot-link constraints $\mathcal{CL}$)
1. Let $C_1, \ldots, C_k$ be the initial cluster centres.
2. For each point $x_i \in \mathcal{X}$, assign it to the closest cluster $C_j$ such that violate-constraints($x_i, C_j, \mathcal{ML}, \mathcal{CL}$) is false. If no such cluster exists, fail (return $\{\}$).
3. For each cluster $C_i$, update its centre by averaging all of the points $x_j$ assigned to it.
4. Iterate between (2) and (3) until convergence.
5. Return $\{C_1, \ldots, C_k\}$.
Procedure: violate-constraints (data point $x$, cluster $C$, must-link constraints $\mathcal{ML}$, cannot-link constraints $\mathcal{CL}$)
1. For each $(x, x_=) \in \mathcal{ML}$: If $x_= \notin C$, return true.
2. For each $(x, x_{\neq}) \in \mathcal{CL}$: If $x_{\neq} \in C$, return true.
3. Otherwise, return false.
```

or soft constraints (SML, SCL), where violations are permitted with associated penalties. The algorithm consists of five sequential steps: preprocessing, initialisation, assignment, update, and post-processing.

Prior to the algorithm's execution, the data is arranged as a weighted undirected graph $G = (V, E)$, where the vertices V correspond to the objects, and the edges E denote the constraints, categorised into four distinct groups: E^{ML} for hard ML, E^{CL} for hard CL, E^{SML} for soft ML, and E^{SCL} for soft CL. Notably, edges representing soft constraints are assigned weights denoted by a confidence value w_{ij}

In the preprocessing phase, the graph $G = (V, E)$ undergoes a transformation into another weighted undirected graph $G' = (V', E')$. This transformation involves contracting all edges $(i, j) \in E^{ML}$, merging nodes connected by hard ML constraints, and adjusting edges to reflect hard CL constraints, along with any remaining soft ML and CL constraints.

In the initialisation step, the initial positions of the k cluster centres are established, offering two distinct methods: either a random selection of points or the adoption of the K-Means++ algorithm introduced by Arthur et al. [16].

During the assignment step, every node in the graph $G' = (V', E')$ is allocated to one of the K clusters, aiming to minimise the total distance between nodes and their respective centres while adhering to both hard and soft pairwise constraints.

Following the assignment step, the positions of the cluster centres are adjusted based on the node assignments from the preceding step, a process iterated as long as there is potential for decreasing the objective function value. The assignment with the most favourable objective value upon termination is forwarded to the postprocessing step. Here, the labels of the graph $G' = (V', E')$ are remapped to the original representation $G = (V, E)$ and returned as the final assignment.

Metrics

The metrics used in this study are widely recognised within the clustering community for evaluating the similarity between clustering results. Their design enables informative comparison between a clustering solution and a ground truth partition by quantifying their level of agreement. Both NMI and the ARI are capable of performing this task. While they are both similarity metrics, they differ in their underlying formulations and evaluation focus: ARI emphasises precise pairwise agreement between partitions, whereas NMI captures the overall structural similarity between clusterings. The specific formulations of each metric are outlined below.

- Normalised Mutual Information (NMI) [128]: NMI is rooted in information theory and is particularly valuable in scenarios where cluster labels themselves are arbitrary, and only the grouping structure is informative. At its core lies Mutual Information (MI), which measures how much knowing the cluster assignment in one clustering reduces uncertainty about the assignment in the other. A high MI value indicates that the two clusterings share substantial information, signifying a strong similarity. However, because MI is unbounded and sensitive to the number of clusters, it is normalised to allow fair comparisons across different datasets and clustering configurations.

  Consider two clustering results $C_1 = \{C_{1,1}, C_{1,2}, \ldots, C_{1,K}\}$ and $C_2 = \{C_{2,1}, C_{2,2}, \ldots, C_{2,L}\}$, where $C_{1,k}$ and $C_{2,l}$ denote individual clusters in clusterings C_1 and C_2, respectively. Let N be the total number of data points, n_{kl} the number of data points shared between clusters $C_{1,k}$ and $C_{2,l}$, $n_{k\cdot}$ the number of data points in cluster $C_{1,k}$, and $n_{\cdot l}$ the number in cluster $C_{2,l}$.

The mutual information, which quantifies the shared information between the two clusterings, is given by equation 5.1:

$$I(C_1; C_2) = \sum_{k=1}^{K} \sum_{l=1}^{L} \frac{n_{kl}}{N} \cdot \log \left(\frac{N \cdot n_{kl}}{n_{k\cdot} \cdot n_{\cdot l}} \right) \tag{5.1}$$

The entropies of C_1 and C_2, representing the uncertainty within each clustering, are calculated using equations 5.2:

$$H(C_1) = -\sum_{k=1}^{K} \frac{n_{k\cdot}}{N} \cdot \log \left(\frac{n_{k\cdot}}{N} \right) \quad \text{and} \quad H(C_2) = -\sum_{l=1}^{L} \frac{n_{\cdot l}}{N} \cdot \log \left(\frac{n_{\cdot l}}{N} \right) \tag{5.2}$$

Finally, NMI, which adjusts for cluster size and scales the mutual information between 0 and 1, is computed using equation 5.3:

$$\mathrm{NMI}(C_1, C_2) = \frac{2 \cdot I(C_1; C_2)}{H(C_1) + H(C_2)} \tag{5.3}$$

- Adjusted Rand Index (ARI): ARI improves upon the original Rand Index by accounting for the agreement that could occur purely by chance, resulting in a more stable and interpretable evaluation metric. It operates by considering all possible pairs of data points and assessing whether they are assigned to the same or different clusters across both clustering solutions. An ARI value of 1 indicates perfect agreement, 0 reflects the level of similarity expected by random assignment, and negative values suggest less agreement than would be expected by chance. The ARI is symmetric and invariant to permutations of cluster labels, making it particularly well-suited for evaluating unsupervised clustering outcomes against a known ground truth or alternative clustering result.

The ARI is computed from the contingency table of the two clusterings. Let n_{kl} denote the number of elements shared between cluster $C_{1,k}$ in clustering C_1 and cluster $C_{2,l}$ in clustering C_2, $c_{1k} = \sum_l n_{kl}$, and $c_{2l} = \sum_k n_{kl}$. Let N be the total number of data points. The ARI is defined with equation 5.4:

$$\text{ARI} = \frac{\sum_{i,j}\binom{n_{kl}}{2} - \left[\sum_k \binom{c_{1k}}{2} \sum_l \binom{c_{2l}}{2} \Big/ \binom{N}{2}\right]}{\frac{1}{2}\left[\sum_k \binom{c_{1k}}{2} + \sum_l \binom{c_{2l}}{2}\right] - \left[\sum_k \binom{c_{1k}}{2} \sum_l \binom{c_{2l}}{2} \Big/ \binom{N}{2}\right]} \tag{5.4}$$

This formulation adjusts the raw agreement score by subtracting the expected agreement due to random chance and normalising by the maximum possible agreement.

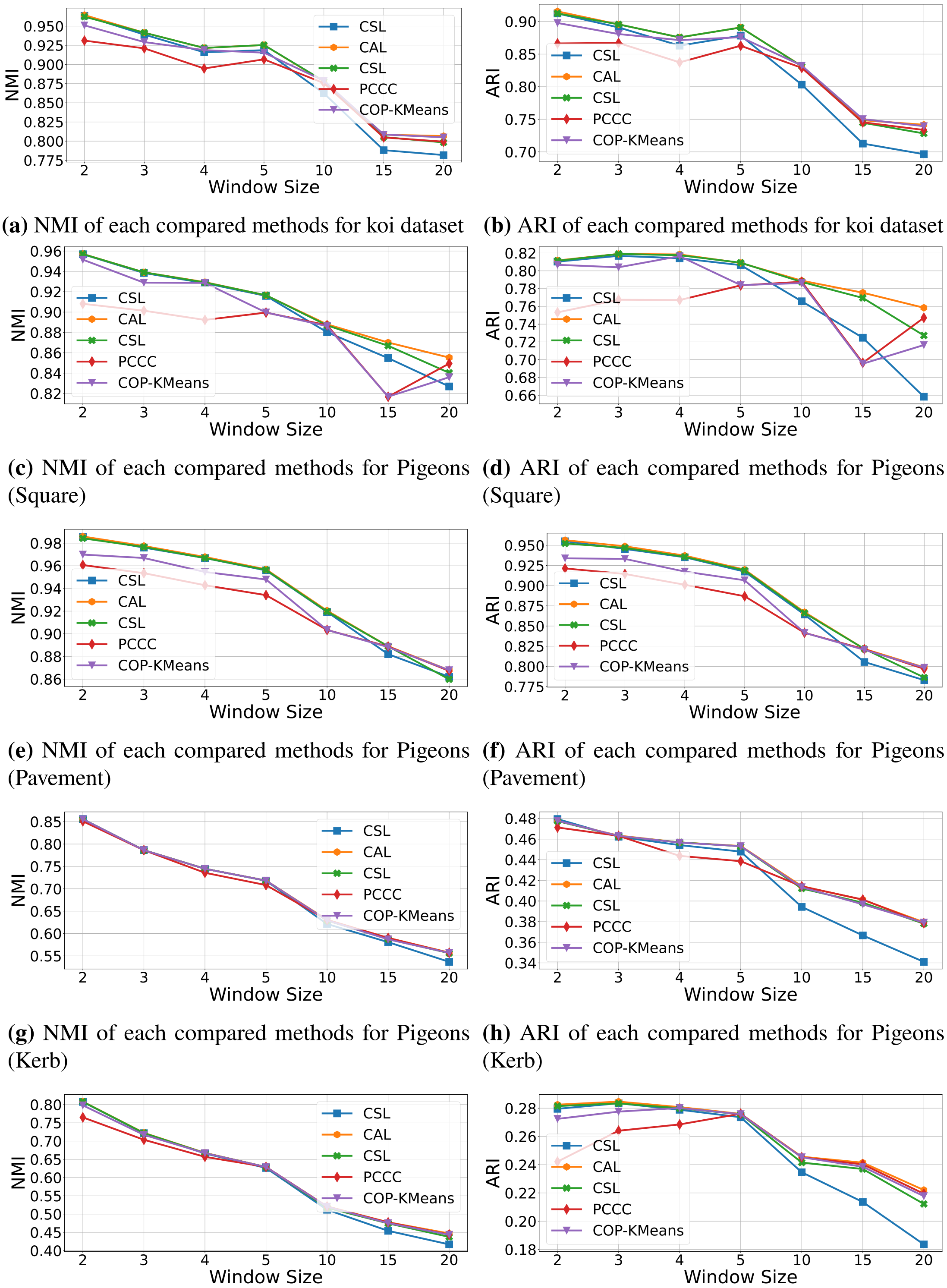
Experimental Protocol

Each video is segmented into windows, where a window comprises a sequence of WS consecutive frames. Our data sets contain the BBs, their description as points in a multidimensional feature space, as well as the label (identity) for each bounding box. Constraints are generated for each window using the protocol outlined in Section 5.1.2. Once the constraints are established, each semi-supervised clustering method is applied to the window to create a partition. Then, we calculate the NMI and ARI between the resulting partition and the ground truth labels. Finally, we compute the average NMI and ARI values across all windows of size $WS \in [2, 3, 4, 5, 10, 15, 20]$ for each clustering method and dataset.

Results

The results of the experiment shown in Figure 5.8 indicate that reducing the window size significantly lessens the complexity of the problem, which in turn allows all the evaluated methods to perform more efficiently. By simplifying the task, the environment becomes more manageable, enabling the algorithms to detect underlying structures with improved clarity and accuracy.

At smaller window sizes, hierarchical clustering methods exhibit superior performance when compared to their centroid-based counterparts. This superiority is reflected not only in the quality of the clustering structure—particularly due to the elongated,



(a) NMI of each compared methods for koi dataset

(b) ARI of each compared methods for koi dataset

(c) NMI of each compared methods for Pigeons (Square)

(d) ARI of each compared methods for Pigeons (Square)

(e) NMI of each compared methods for Pigeons (Pavement)

(f) ARI of each compared methods for Pigeons (Pavement)

(g) NMI of each compared methods for Pigeons (Kerb)

(h) ARI of each compared methods for Pigeons (Kerb)

(i) NMI of each compared methods for Pigs

(j) ARI of each compared methods for Pigs

Figure 5.8: Illustrates the NMI and ARI values for each window size, dataset, and clustering method. Each clustering method is distinguished by a unique colour and marker, as indicated in the plot legend.

string-like clusters observed in these tighter windows—but also in the alignment between the clustering results and the known ground truth. The data characteristics within smaller windows appear to align well with the assumptions underpinning hierarchical techniques.

However, as the window size increases, the complexity and diversity of the problem also expand. In these scenarios, the distinction between clustering approaches becomes increasingly ambiguous. This trend suggests that as additional data are incorporated into the window, the resulting cluster structures exhibit greater heterogeneity in form, thereby reducing the clarity in selecting an appropriate clustering method.

Overall, the key insight is that simplifying the problem—specifically by reducing the window size—leads to improved performance across all methods. This highlights the potential effectiveness of adopting an online approach, where data is handled incrementally in smaller, more manageable portions, enabling more consistent and reliable clustering results over time.

It should be noted that this study does not address the linking of consecutive windows, which would provide a more comprehensive evaluation of each clustering approach in a truly online setting. While identities are established independently within each window, deriving an overall performance metric would require the association of these identities across successive windows. This would enable the construction of a consistent set of identities observed throughout the entire video sequence, allowing comparison against the ground truth.

Although tracking and maintaining identities across an entire sequence is a key component of online re-identification, the aim of this study was to evaluate the suitability of different clustering approaches for the types of data structures commonly found in animal video footage. These structures were made more discernible by reducing complexity through the use of smaller window sizes.

5.2 A New Online Constrained Clustering Approach

In this section, a new Real-Time, Species-Invariant (RTSI-ReID) approach is presented for the online re-identification of animals in live video footage. Building upon the findings from the experimental study in Section 5.1, which showed that processing fewer frames at a time improved the performance of all clustering methods evaluated, the logical next step involves analysing the video one frame at a time.

5.2.1 Method Overview

Given a video stream $\mathcal{V} = \{F_1, \ldots, F_T\}$ consisting of T frames — where T may be unbounded — the objective is to re-identify animals appearing in the scene and annotate each frame of $\mathcal{V}$ with BBs and their corresponding identities. To achieve this, we propose a frame-by-frame online constrained clustering approach that addresses the intrinsic difficulties associated with online clustering and animal re-identification, as well as each of the challenges outlined in Section 3.3.

With the continuous advancement of object detection techniques and feature extraction methods, it is essential to develop a solution capable of adapting to these developments. As such, the initial stages of the animal re-identification process are considered external to the real-time algorithm introduced in this section, thereby allowing users the flexibility to select their preferred object detector and feature representation technique.

As each frame F_t is received, the selected object detector is first applied to identify and return the BBs of all detected objects within the frame. Each bounding box is represented as a four-tuple of pixel coordinates within the frame, denoted $B_i = \langle x_{\text{top}}, y_{\text{top}}, w, h \rangle$. These BBs are then transformed into the feature space using the chosen feature extraction method. Once all BBs and their corresponding feature representations $\mathbf{x}_i$ have been obtained, they are passed into the proposed algorithm. The resulting pairs of BBs and features, denoted $D_{t,i} = \langle B_i, \mathbf{x}_i \rangle$, are referred to as *detections*.

When developing an online clustering solution to process a potentially unbounded stream of data, careful consideration must be given to how the data is stored. Retaining all data received is not only impractical but also impossible, as no system can store an infinite volume of information. To address this limitation, protocols have been devised to summarise cluster information using a range of statistical measures, sometimes referred to as the *cluster footprint* [25]. Common examples include the cluster centroid, covariance matrix, count of assigned points, and temporal statistics such as the timestamp of the most recent update.

By maintaining a cluster footprint for each cluster, based on the data observed up to time t, the algorithm is able to discard individual data points from memory once they have been processed. This approach supports a memory-efficient solution suitable for handling continuous and unbounded data streams. However, it is crucial to carefully select which summary statistics are retained, as they directly impact the algorithm's ability to perform effectively without access to the full historical data. Accordingly, we represent each cluster C_i using the following statistics:

$$C_i = \langle \mu_i, \Sigma_i, B_i, n_i, \delta_i \rangle, \quad i = 1, \ldots, K,$$

where, μ_i denotes the multivariate mean of the cluster, Σ_i represents the covariance matrix, B_i corresponds to the bounding box of the most recently added object in the cluster, n_i indicates the number of objects within the cluster, δ_i is the number of frames since the last object was added to the cluster, and c is the current number of clusters. We denote the collection of all C_i by C.

Storing the multivariate mean μ_i of a cluster enables us to monitor the cluster's locality within the feature space and provides a reference point for comparison with incoming data. The covariance matrix Σ_i defines the relationship between the points within the cluster and, in the context of a multivariate normal distribution, allows us to estimate the convex bounding region of the cluster. The covariance matrix also enables us to estimate the likelihood of a point belonging to a given cluster, assuming a multivariate normal distribution. One of the most challenging aspects of online animal re-identification is determining whether a newly observed object belongs to an existing cluster (identity) or represents a new one. By leveraging both μ_i and Σ_i, we aim to distinguish between existing and novel clusters.

As each object is removed from memory once it has been processed, it is vital that we store the bounding box of the last object, B_i, to be added to a cluster C_i, along with a time stamp, δ_i, to determine how long ago this object was added. The purpose of these summary statistics is to enable the algorithm to calculate the instance-level ML constraint between the objects in the current frame F_t and the existing clusters. By ensuring that the time stamp of a cluster C_i indicates that the last time an object was added to that cluster was in the previous frame, we can then apply the protocol outlined in Section 5.1.2 between B_i and each detection in F_t, denoted by $\mathcal{D}_t = \{D_{t,1}, \ldots, D_{t,M}\}$, where M detections have been identified.

The RTSI-ReID algorithm operates on video data in a sequential, frame-by-frame manner, with the objective of assigning a consistent identity label to each object detected within a given frame F_t. The algorithm is initialised with an empty set of clusters, $C = \emptyset$, and remains idle until a frame containing at least one detection is encountered. Upon detection, each of the M detections in the frame initiates a new cluster $C_i = \langle \mu_i, \Sigma_i, B_i, n_i, \delta_i \rangle$: the object's feature representation serves as the cluster mean μ_i, the covariance matrix Σ_i is set to the identity matrix, $n_i = 1$, B_i corresponds to the bounding box of the detection, and the timestamp $\delta_i = 0$, for each $i = 1, \ldots, M$. Each new cluster C_i is then added to C, i.e. $C = C \cup \{C_i\}$ such that, C become the collection of cluster footprints $\{C_1, \ldots, C_M\}$.

The process begins by evaluating ML constraints between all BBs B_i associated with detection from the previous frame $\mathcal{D}_{t-1} = \{D_{t-1,1}, \ldots, D_{t-1,L}\}$ and the detections in the current frame. $\mathcal{D}_t = \{D_{t,1}, \ldots, D_{t,M}\}$. This evaluation yields $L \times M$ IoU values. An IoU score exceeding the predefined threshold $\tau_{ML} = 0.6$ indicates a potential ML constraint, a threshold informed by IoU guidelines, which recommend values between 0.5 and 0.7 as indicative of sufficient overlap in object detection tasks. However, given that the present context involves moving objects, the overlap requirement is relaxed to avoid overly restrictive matching. In cases where multiple ML constraints are identified for a single detection, the association with the highest IoU value is selected, ensuring that at most one ML constraint is assigned to each detection.

Detections that satisfy the ML criterion are assigned to their corresponding clusters, which are subsequently updated. Any remaining detections—those not matched via ML constraints—must either be allocated to an unused cluster or used to initialise a new one, depending on their similarity to existing clusters. To facilitate this, the algorithm maintains a set of linked clusters $C^L = \emptyset$ at the onset of each frame's processing, which records clusters that have already been updated. When a detection is assigned to a cluster C_i, the identification number i of the cluster is added to the set C^L, i.e.,

$C^L = C^L \cup \{i\}$. This mechanism ensures that no more than one detection is assigned to any individual cluster per frame, thereby upholding the CL constraint.

Following the ML assignment phase, any detections that remain unassigned are either matched to a cluster not yet updated in the current frame, denoted $C^U = C \setminus C_{C^L}$, where C_{C^L} represents the set of cluster footprints corresponding to the identification numbers in C^L, or used to initialise new clusters. To distinguish between previously seen identities and novel ones, the log-likelihood of each unassigned detection belonging to each candidate cluster $C_i \in C^U$ is computed under a multivariate normal distribution model, using the following expression:

$$\log \mathcal{L}(\mathbf{x} \mid C_i) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(\mathbf{x} - \mu_i)^\top \Sigma_i^{-1}(\mathbf{x} - \mu_i), \tag{5.5}$$

where $\mathbf{x}$ denotes the feature representation of the detection, and d is the dimensionality of $\mathbf{x}$ which, in this instance, equals 54 since RGB features are employed.

The Hungarian algorithm [108] is then applied to the negative log-likelihood matrix $-\mathbf{M}^L$, where $\mathbf{M}^L_{ji} = \log \mathcal{L}(\mathbf{x}_j \mid C_i)$, to determine the set of optimal assignment of detections to clusters, denote this set by $\mathcal{H}$. A detection $D_{t,j}$ is assigned to cluster C_i if the log-likelihood $\log \mathcal{L}(\mathbf{x}_j \mid C_i)$ exceeds a predefined threshold β. Conversely, if $\log \mathcal{L}(\mathbf{x}_j \mid C_i) < \beta$, the detection is considered to represent a new identity and is used to initialise a new cluster.

Whenever a detection is added to an existing cluster C_i, the cluster's summary statistics are updated accordingly. The updated parameters are computed as follows:

$$\mu_i = (1 - \alpha) \cdot \mu_i + \alpha \cdot \mathbf{x}, \tag{5.6}$$

where $\alpha \in [0, 1]$ is a novelty parameter that determines the degree to which the new observation influences the cluster mean. A value of $\alpha = 0$ results in no change to the mean, while $\alpha = 1$ fully replaces it with the new feature vector $\mathbf{x}_j$.

$$\Sigma_i = I, \tag{5.7}$$

$$B_i = B_{\mathbf{x}}, \quad n_i = n_i + 1, \quad \delta_i = 0. \tag{5.8}$$

Equation 5.7 denotes the reduction of the covariance matrix Σ_i to the identity matrix I. This simplification entails that the evaluation of $\log \mathcal{L}(\mathbf{x} \mid C_i)$ is likewise simplified, as inter-feature covariance is neglected in the computation. As a result, Equation 5.5 reduces to a formulation equivalent to the Euclidean distance with an associated penalty, which may be expressed as follows:

$$\log \mathcal{L}(\mathbf{x} \mid C_i) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}(\mathbf{x} - \mu_i)^2, \tag{5.9}$$

Furthermore, Equation 5.8 specifies the updates to B_i, n_i, and δ_i within the summary statistics of a cluster when a new point is incorporated. The bounding box B_i is updated to correspond to the bounding box of the newly added detection $D_{t,j}$, the number of points n_i in the cluster is increased by one, and the time variable δ_i, representing the interval since the previous object was added, is reset to zero.

These updates ensure that the cluster representation remains current and reflective of the most recent assignment, while also preserving computational efficiency suitable for online, streaming data scenarios. Note that, in the basic version of our RTSI-ReID, we have chosen to update only the cluster mean and not the covariance matrix. Also, note that no prior probabilities are included in the model. This reflects the fact that we expect that animals will appear in the video, in and out of camera view, in an unpredictable manner.

The RTSI-ReID algorithm for processing one frame is detailed in Algorithm 1 and continued in Algorithm 2.

Algorithm 1 RTSI-ReID

Input: Current cluster footprints (C)
Detections in the current frame $\mathcal{D}_t = \{D_{t,1}, \dots, D_{t,M}\}$
Parameters α (novelty) and β (acceptance log-likelihood)

Output: The updated cluster footprints C and the labels $y_1, \dots, y_M$ of the detections.

```
Initialise liked clusters $C^L = \emptyset$
Initialise the number of clusters $k = |C|$
if Current clusters is empty $C = \emptyset$ then
    Initialise $C$ with detections $\mathcal{D}_t$.
else
    $\mathcal{D}^{ML}$ ← Calculate which of the detections in $\mathcal{D}_t$ that have ML constraints.
    for each detection $D_{t,j}$ in the set $\mathcal{D}^{ML}$ do
        Identify the cluster $C_i$ that $D_{t,j}$ is linked with.
        Add $D_{t,j}$ to cluster $C_i$ and Update the cluster $C_i$ footprint using equations (5.6)-(5.8) and novelty parameter $\alpha$.
        Add the identifier of cluster $C_i$ to linked clusters $C^L$ i.e. $C^L \leftarrow C^L \cup \{i\}$.
        Assign the cluster label $i$ to the detection $D_{t,j}$ i.e. $y_j \leftarrow i$.
    end for
    $C_{C^L}$ ← The cluster footprints $C_i$ of clusters identified in $\mathbf{C}^L$
    $C^U$ ← Calculate the set of clusters which have not been added to $C \setminus C_{C^L}$
    if There are clusters that have not been added to i.e. $C^U \neq \emptyset$ & there are detections not yet assigned to a cluster i.e. $\mathcal{D}_t \setminus \mathcal{D}^{ML} \neq \emptyset$ then
        $\mathbf{M}^L_{|C^U|,|\mathcal{D}_t \setminus \mathcal{D}^{ML}|}$ ← Initialise the log-likelihood matrix to all zeros
        for each detection $D_{t,j}$ without a ML constraint i.e. $D_{t,j} \in \mathcal{D}_t \setminus \mathcal{D}^{ML}$ do
            for each unassigned cluster $C_i \in C^U$ do
                $\mathbf{M}^L_{ji}$ ← calculate the log-likelihood of detection $D_{t,j}$ belonging to cluster $C_i$ using equation 5.5 $\log \mathcal{L}(D_{t,j}|C_i)$ and update the log-likelihood matrix $\mathbf{M}^L$.
            end for
        end for
        $\mathcal{H}$ ← Calculate the optimal pair assignment of unassigned detection to unlinked clusters by applying the Hungarian algorithm on $-\mathbf{M}^L$
        for each pair of detections and clusters in $\mathcal{H}$ i.e. $(D_{t,v}, C_w) \in \mathcal{H}$ do
            if The log-likelihood between the pair exceeds the acceptance threshold $\beta$ i.e. $\mathbf{M}^L_{v,w} > \beta$ then
                Add $D_{t,v}$ to cluster $C_w$ and Update the cluster $C_w$ footprint using equations (5.6)-(5.8) and novelty parameter $\alpha$.
                Assign the cluster labels $w$ to the detection i.e. $y_v \leftarrow w$.
            else if the log-likelihood between the pair is below the acceptance threshold $\beta$ i.e. $\mathbf{M}^L_{v,w} < \beta$
                Initialise a new cluster $C_{k+1}$ with detection $D_{t,v}$.
                Update the current clusters to include the newly initialised cluster i.e. $C \leftarrow C \cup \{C_{k+1}\}$.
                Assign the cluster labels $k + 1$ to the detection i.e. $y_v \leftarrow k + 1$.
                Increment the number of clusters i.e. $k \leftarrow k + 1$
            end if
        end for
```

Algorithm 2 RTSI-ReID Continued

```
    else If there are detections waiting to be assigned i.e. $\mathcal{D}_t \setminus \mathcal{D}^{ML} \neq \emptyset$ & there are no clusters left i.e. $C^U = \emptyset$
        for Each remaining detection $D_{t,j} \in \mathcal{D}_t \setminus \mathcal{D}^{ML}$ do
            Initialise a new cluster $C_{k+1}$ with detection $D_{t,j}$.
            Update the current clusters to include the newly initialised cluster $C \leftarrow C \cup \{C_{k+1}\}$.
            Assign the cluster labels $k + 1$ to the detection i.e. $y_j \leftarrow k + 1$.
            Increment the number of clusters $k \leftarrow k + 1$
        end for
    end if
end if
Return $y_1, \dots, y_k$ and $C$.
```

5.2.2 Methods included in the experiment

- **RTSI-ReID.** The proposed method.

- **ML-ReID. (BASELINE)** This is a variant of the proposed method that utilises only the ML constraints to form chains, with each chain corresponding to an individual cluster. Any point that does not satisfy an ML constraint with an existing chain initiates a new one. Consequently, this approach constructs ML chains and assigns each to a cluster, thereby evaluating performance based solely on ML constraints.

- **IKM.** Incremental K-Means [155] assigns new observations to the closest cluster and moves the centre of the cluster towards the new observation. The amount by which the cluster centre is moved towards the new observation is determined though a parameter *halflife* .

- **DBSTREAM.** DBSTREAM [81] is the first micro-cluster-based online clustering component that explicitly captures the density between micro-clusters via a shared density graph. The density information in the graph is then exploited for re-clustering based on density between adjacent micro clusters to form macro clusters of arbitrary shapes.

- **KG.** Kulshreshtha et al. [110] introduce a two-stage online clustering approach for person re-identification, achieving results comparable to or surpassing state-of-the-art offline and online methods. The first stage focuses on generating facial tracks from a given window. By employing a standard face detector alongside a deep feature extractor, they construct these tracks using intersection over union (IoU) calculations between bounding box detections in consecutive frames and by minimising feature distance between corresponding representations.

In the second stage, the method clusters tracks sequentially as they appear, aiming to identify previously seen identities and group the tracks accordingly. This process relies on three matrices:

- Temporal Constraint Matrix (**Q**) – Determines the duration of each face track and detects overlaps between different tracks, enforcing ML constraints to prevent incorrect clustering.

- Similarity Matrix (**S**) – Measures the similarity between face tracks and cluster centres within a given window.

- Weight Matrix (**W**) – Initially set to all ones and later updated using the temporal constraint matrix (**Q**).

These matrices collectively facilitate the clustering process, enhancing the accuracy of identity re-identification throughout the video. By computing the element-wise product of **S** and **W** and assessing whether the maximum value exceeds a predefined threshold τ, the algorithm determines whether a track should be assigned to an existing cluster or classified as a new identity.

5.2.3 Experimental Protocol

In order to test the true capability of the proposed method, we assumed that a perfect object detector is available. This means that, in each frame, the BBs for the objects are the ground truth ones (human annotation). This is kept for all methods used in the comparison.

For the evaluation of the IKM method, each frame was processed sequentially, with individual processing of each detection found within the frames returning the corresponding label. To maximise the algorithm's potential for success, we used the true number of clusters K while varying the $halflife$ parameter $halflife = \{0.1, 0.2, \ldots, 0.9\}$ to evaluate the performance under different amounts of concept drift. Resulting in 9 ARI values for each dataset.

To evaluate the performance of the DBSTREAM algorithm, an identical processing protocol was followed: each frame was handled sequentially, and every detection within each frame was individually processed to assign a corresponding label. Only two parameters were varied — the clustering threshold $\in \{10, 11, ..., 15\}$, which defines the required density within a user-defined radius around the cluster centre, and the fading factor $\in \{0.1, 0.2, ..., 0.9\}$, which determines how much influence past data has on the current clustering state. All other parameters were held constant, with the clean-up interval set to 4, the intersection factor at 0.05, and the minimum weight fixed at 1.

To evaluate the performance of KG, each video was segmented into windows of varying sizes, $WS = \{2, 3, ..., 10\}$ (9 values). Additionally, the user-defined parameter τ was varied as $\tau = \{0.5, 1.0, ..., 3.5\}$ (7 values), a range influenced by the similarity values within the original algorithm. Once the window segments were prepared, the KG algorithm was applied on each set of windows of a given size in sequential order. The KG algorithm returns labels for all objects in the entire video which are then compared with the ground truth using the ARI. This process produced 63 individual results for each dataset.

For all remaining methods, each dataset was divided into individual frames, where the corresponding detections (BBs and extracted feature representations) within each frame were batched and passed to the algorithm's update method. With each frame, the clusters were updated and labels were returned for each detection. After processing all frames of a video, the assigned labels were compared to the ground truth labels using the ARI, measuring the similarity between the two sets of labels. The frame processing time was also recorded to evaluate the speed of the proposed method and ensure the capability of 60fps processing speeds.

For the proposed method and its variants, each video was processed for each combination of $\alpha = \{0.1, 0.2, ..., 0.9\}$ and $\beta = \{0, -100, ..., -1500\}$, resulting in a total of 144 individual results for each dataset. The range of values for β was decided upon based on some pilot experiments.

5.2.4 Results

Table 5.2 demonstrates that the proposed method outperforms all other approaches in the task of animal re-identification from video. This strong performance may be largely attributed to two principal heuristics: frame-by-frame processing and the autonomous generation of instance-level constraints by the algorithm. The effectiveness of these constraints is further underscored by the performance of the baseline method ML-ReID, which exceeds that of all other competing approaches despite relying exclusively on instance-level constraints without additional processing.

Table 5.2: This table presents the highest ARI attained by each of our competitors, our proposed method, and the baseline variant across all five datasets used in the experiment. The top-performing method for each dataset is highlighted in green.

		Method			
	ML-ReID	IKM	DBSTREAM	KG	RTSI-ReID
Koi Fish	0.6137	0.0000	0.1174	0.0348	0.7440
Pigeons (Square)	0.4497	0.0058	0.1783	0.0078	0.7706
Pigeons (Pavement)	0.5056	0.0924	0.0813	0.0120	0.5207
Pigeons (Kerb)	0.5077	0.0642	0.0633	0.0086	0.6829
Pigs	0.4284	0.0311	0.0707	0.0037	0.8078

The frame-by-frame approach enables the method to fully exploit both ML and CL constraints. This strategy ensures the continuity of ML chains across successive frames, while simultaneously supporting the consistent tracking and enforcement of CL constraints.

Figure 5.9 illustrates the average ARI achieved by IKM across all five datasets in our experiment, evaluated over varying values of the halflife parameter. The figure shows that the method performs best with a halflife value of 0.7, suggesting that a greater degree of concept drift is more suitable for our datasets. Nonetheless, the overall performance of IKM remains poor for the given task. We attribute this to the complex structures and highly intertwined clusters within our data. As IKM inherently favours spherical cluster formations, it struggles to adequately capture the intricacies present within the datasets.

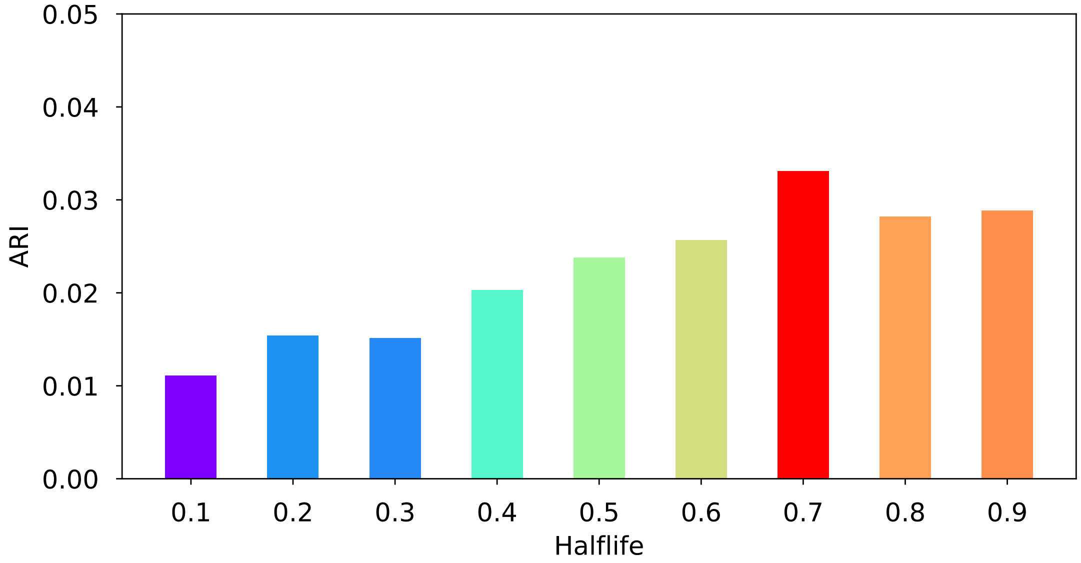


Figure 5.9: Illustrates the average ARI across all five datasets for IKM across different values of the halflife parameter. Each halflife value is colour-coded using a gradient from red to purple, with red indicating the best performance and purple the worst.

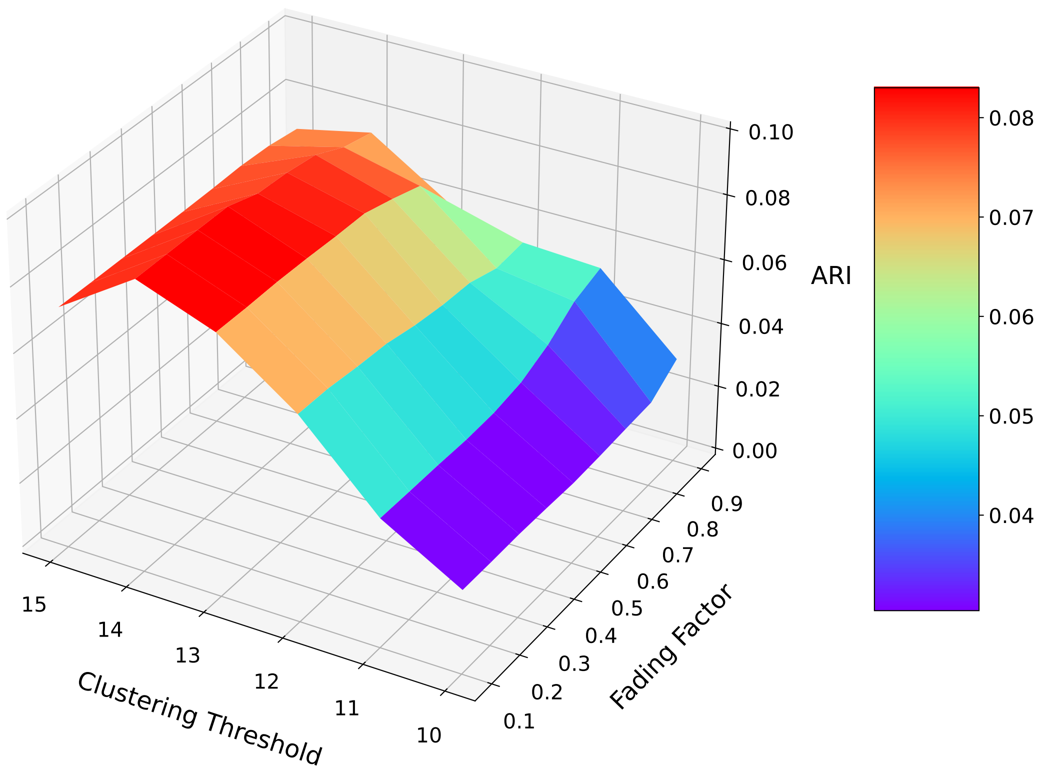


Figure 5.10: Illustrates the average ARI across all five datasets for DBSTREAM over all combinations of the clustering threshold and fading factor parameters. The surface is colour-coded to represent performance, with values mapped to the corresponding scale shown in the colour bar.

Figure 5.10 presents the average performance of DBSTREAM across all five datasets, evaluated over a range of clustering threshold and fading factor values. The clustering threshold determines the required density around a cluster centre within a user-defined radius, while the fading factor regulates the influence of historical data. The plot indicates that the method performs best with a clustering threshold of 14 and lower fading factor values.

However, the overall results suggest that DBSTREAM is not well-suited to the task of animal re-identification from video. This limitation is likely attributable to its density-based micro-clustering approach. While DBSTREAM can capture complex structures by forming and merging micro-clusters based on shared density, it struggles with the overlapping clusters and substantial concept drift characteristic of our datasets. Consequently, it fails to consistently distinguish between identities and has difficulty re-identifying individuals once they reappear in a different region of the feature space.

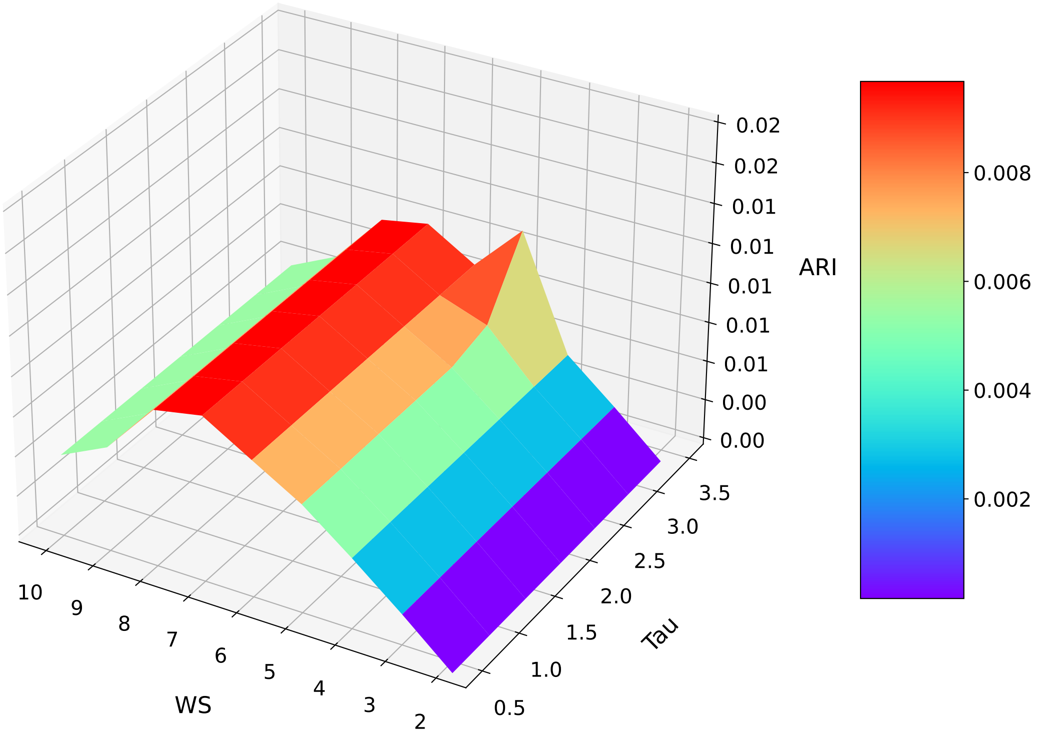


Figure 5.11: Illustrates the Average ARI across all 5 datasets for all combinations of parameters Window size (WS) and user defined parameter τ for KG. The surface is colour-coded to represent performance, with values mapped to the corresponding scale shown in the colour bar.

Figure 5.11 displays the average ARI achieved by KG across all five datasets. Despite its demonstrated success in facial re-identification tasks, the results indicate that KG is not well-suited to species-invariant animal re-identification, irrespective of the window size employed. This limitation may be attributed to two principal aspects of the method: its reliance on deep features and its windowed processing approach.

KG was developed with deep feature extraction at its core—an approach that performs well in human facial recognition, where discriminative features can be effectively captured by deep neural networks, leading to well-separated clusters in the feature space. However, such deep, distinctive features are not readily available in our datasets, presenting challenges that KG is unable to address effectively.

Moreover, the method's windowed processing paradigm may further constrain its performance. Although KG incorporates instance-level constraints akin to those in our proposed method, processing data in fixed-size windows can disrupt the continuity of detections. When the actual sequence of related detections spans beyond the window boundary, the constraints may fail to establish a complete linkage, thereby reducing overall re-identification accuracy.

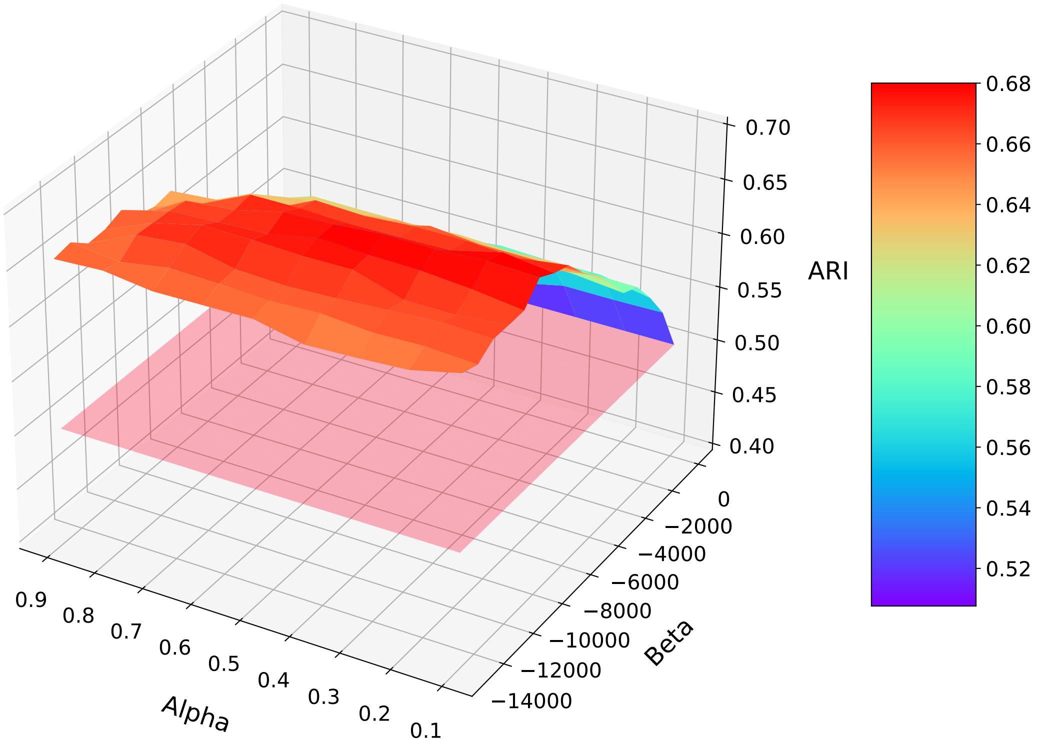


Figure 5.12: Illustrates the average ARI across all five datasets for RTSI-ReID over all combinations of the parameters α and β. The surface is colour-coded to represent performance, with values mapped to the corresponding scale in the colour bar. The red surface in the plot denotes the average ARI achieved by the baseline ML-ReID method.

Figure 5.12 presents the ARI values comparing the labels generated by RTSI-ReID against the ground truth. Each value corresponds to a unique combination of α and β. The red surface in the plot represents the performance of the ML-ReID variant, which serves as a baseline relying solely on instance-level constraints. It is evident from the figure that the incorporation of likelihood statistics—alongside spatio-temporal instance-level constraints—enhances the method's capacity to discern whether a new data point belongs to an existing cluster or represents a previously unseen identity. The plot also indicates that a lower β value, corresponding to a more lenient acceptance threshold, results in improved clustering outcomes. Similarly, a lower α value, which governs the regulation of concept drift, enables the method to more effectively track identities across broader regions of the feature space.

However, clustering accuracy alone is insufficient for the development of a real-time, species-invariant clustering method. Figure 5.13 illustrates the frame processing time of the proposed method for each combination of α and β. The red surface in the plot denotes the minimum required processing time to maintain a video playback rate of 60 frames per second; any value below this surface is considered acceptable. It is important to note, however, that these timings do not include the additional processing required for object detection and feature extraction prior to frame-level clustering. While not a comprehensive representation of the total pipeline's processing time, the method demonstrates sufficient efficiency to accommodate these supplementary processes without compromising real-time performance.

5.2.5 Ablation Study

To assess the importance of each individual component within the proposed method, an ablation study was carried out using various configurations of RTSI-ReID. This systematic evaluation enabled a detailed analysis of the contribution made by each element to the overall performance of the model. By comparing different variations, the study provided empirical evidence supporting the necessity and effectiveness of each component in enhancing the method's stability and accuracy.

Methods

- **RTSI-ReID.** The proposed method.

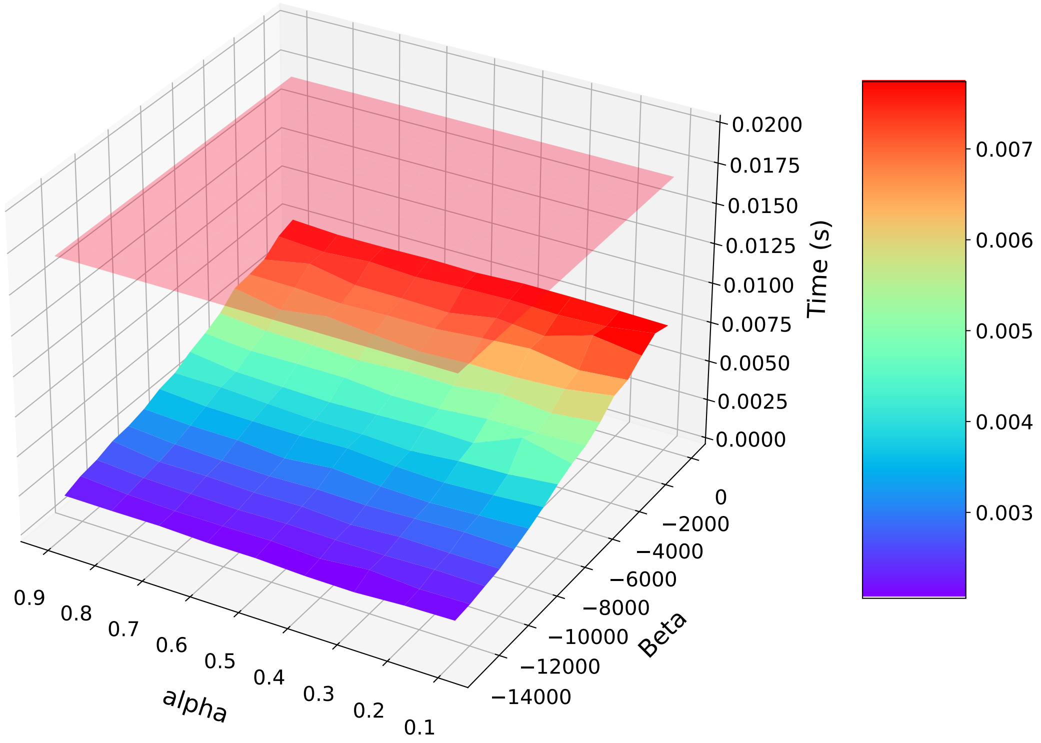


Figure 5.13: This figure displays the Average processing time of a single frame across all 5 datasets for all combinations of parameters α and β used within the experiment. The red surface in the plot indicates the a threshold of $\frac{1}{60}$ i.e. the time required to process a video at 60 fps .

- **ML-ReID. (BASELINE)**

- **LS-ReID.** In this version of RTSI-ReID, instance-level ML constraints are excluded. Instead, only likelihood statistics are used to associate detections with existing clusters or to determine whether a detection represents a previously unseen identity. CL constraints remain consistently integrated into the model. This configuration serves as an ablation study to assess the contribution of ML constraints within RTSI-ReID.

- **RTSI-ReID+.** In this version, we incrementally update the covariance matrices within the cluster footprints as detections are added to them. Instead of equation (5.7) we use:

$$\Sigma_i = \frac{n_i - 1}{n_i} \cdot \Sigma_i + \frac{(\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T}{n_i + 1} \tag{5.10}$$

Table 5.3: This table displays the maximum ARI achieved by each variant of our proposed method on all 5 datasets used in our experiment. The best performing method for each dataset is highlighted in green.

	Method			
	ML-ReID	LS-ReID	RTSI-ReID+	RTSI-ReID
Koi Fish	0.6137	0.4559	0.7466	0.744
Pigeons (Square)	0.4497	0.4316	0.6115	0.7706
Pigeons (Pavement)	0.5056	0.5055	0.5056	0.5207
Pigeons (Kerb)	0.5077	0.5077	0.7086	0.6829
Pigs	0.4284	0.4269	0.6960	0.8078

Results

Table 5.3, together with Figures 5.12, 5.15 and 5.14, presents the results of the ablation study and highlights the respective strengths of methods RTSI-ReID and RTSI-ReID+. However, due to its greater stability, simplicity, and superior average performance, the proposed method, RTSI-ReID, was selected over the more complex and less stable alternative.

Figure 5.14 presents the average ARI scores (across all datasets) for LS-ReID, evaluated across all combinations of the α and β parameters. The red surface represents the baseline ML-ReID. The plot demonstrates that relying solely on likelihood statistics is insufficient for effective species-invariant animal re-identification. Furthermore, it highlights that the baseline ML-ReID consistently outperforms the LS-ReID variant, underscoring the added value of incorporating instance-level constraints.

Figure 5.15 presents the average ARI values (across datasets) for RTSI-ReID+, evaluated across varying values of α and β. The red surface once again represents the baseline performance of ML-ReID. It is evident that RTSI-ReID+ achieves its highest performance with a higher acceptance threshold (β) combined with a medium value of α. Additionally, the results indicate that method RTSI-ReID+ is considerably more sensitive to the choice of α and β, suggesting limitations in the use of a multivariate normal distribution to approximate the complex structures inherent in the data.

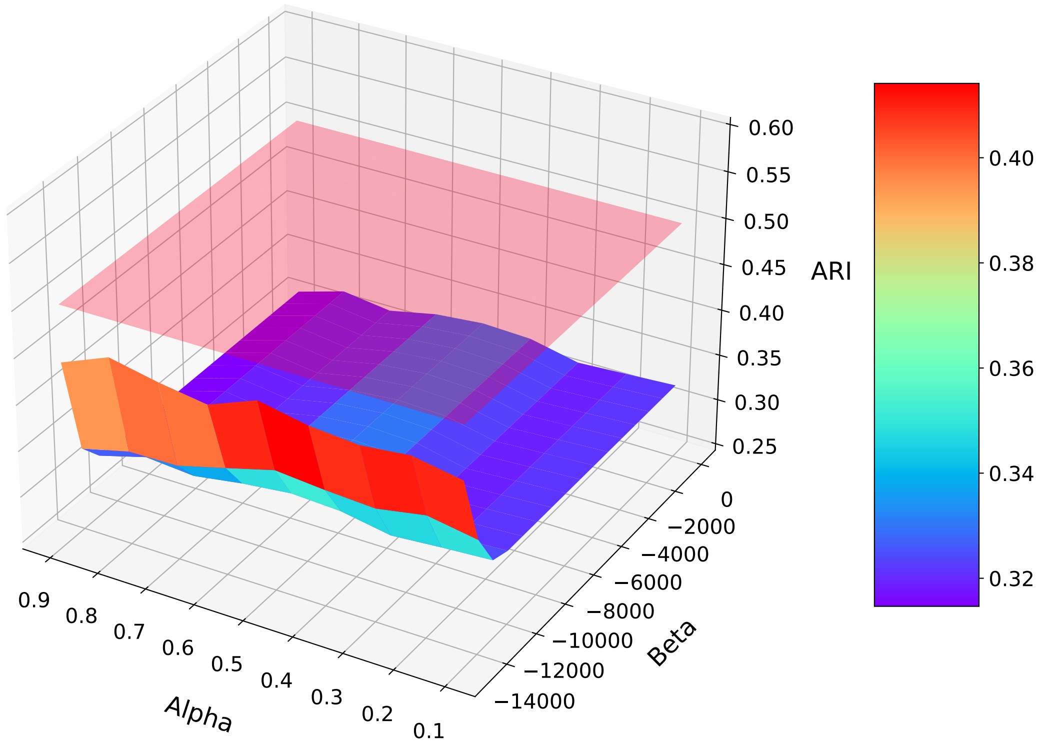


Figure 5.14: Illustrates the average ARI across all five datasets for LS-ReID across all combinations of the parameters α and β. The surface is colour-coded to indicate performance, with values mapped to the corresponding scale in the colour bar. The red surface in the plot represents the average ARI of the baseline ML-ReID method.

Enhancements could be made through more advanced techniques for distinguishing between new and existing identities, as well as by incorporating more detailed feature representations that are effective across a wide range of animal species.

5.3 Summary

This chapter presented a comparative study to determine the most appropriate clustering approach — centroid-based or hierarchical — for animal re-identification, highlighting the superior efficacy of hierarchical clustering methodologies. The experiment also demonstrated that reducing data complexity—specifically by decreasing the size of the window processed by the clustering algorithm—improved the performance of all evaluated methods. This finding suggests that online processing may represent a more effective approach to animal re-identification compared to storing the data and processing it in a batch manner.

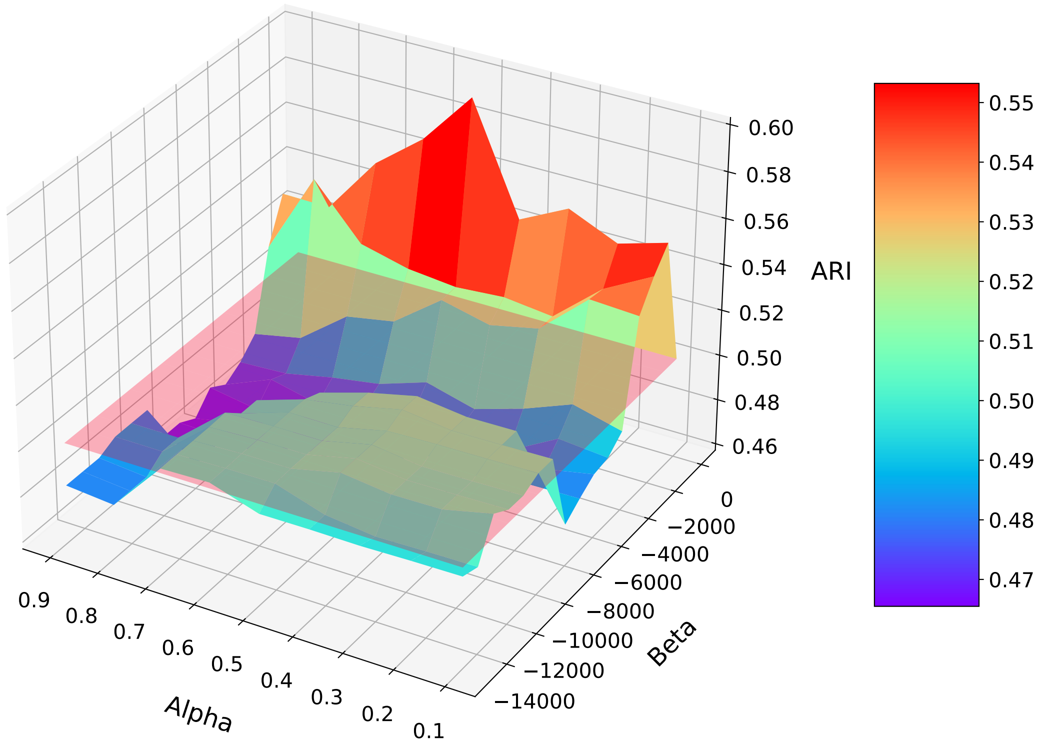


Figure 5.15: Average ARI across all five datasets for RTSI-ReID+ using the parameters α and β employed in the experiment. The surface is colour-coded to represent performance, with values mapped to the corresponding scale in the colour bar. The red surface in the plot denotes the average ARI of the baseline ML-ReID method.

Building on the results of this comparative study, a novel real-time species-invariant constrained clustering method, RTSI-ReID, was proposed, employing a frame-by-frame processing paradigm to capitalise on the reduced data complexity. When compared against other online clustering approaches, including a state-of-the-art facial re-identification clustering method, RTSI-ReID consistently outperformed its competitors. This success is attributed to its frame-by-frame processing strategy and its capability to track and maintain constraints throughout the video stream.

Chapter 6

Offline Animal Re-Identification

Although online methods for animal re-identification enable real-time tracking and monitoring—thereby enhancing livestock management and contributing to wildlife conservation—there are scenarios, such as the analysis of pre-recorded footage, in which the entire dataset is available in advance. In these contexts, the development of stable offline solutions becomes crucial for achieving accurate re-identification, as the absence of real-time restrictions allows for full exploitation of the available data.

Nonetheless, even with access to the complete dataset, designing an effective offline clustering method for animal re-identification remains challenging. As discussed in Section 3.3, the intricate and interconnected structures that emerge across a video sequence pose significant obstacles to the development of reliable offline approaches.

This chapter presents two novel offline constrained clustering methods developed to address the specific challenges of animal re-identification using video data in offline contexts. These methods are intended to tackle the inherent complexities of video-based datasets, such as temporal variation and individual similarity.

Contributions covered in this Chapter

Evaluating a constrained clustering ensemble method for clustering a variety of real and synthetic datasets & Proposing a classification-based clustering method for clustering a range of real world video datasets.

Publications 6 & 7 in Section 1.4

3

6.1 Classification-based clustering (CBC)

The concept of classification-based clustering involves utilising the outputs of object detection and MOT algorithms — specifically, the generated tracks — to construct a partial structural representation of the dataset. Although these tracks may offer an incomplete depiction of the underlying data structures, they can nevertheless serve as a foundation for training a classification model. This model can then be employed to identify and merge tracks based on patterns of misclassification between classes, ultimately facilitating the construction of a complete partition of the dataset.

6.1.1 Methodology

Preliminaries

Let $\mathcal{V} = \{F_1, \ldots, F_T\}$ denote a sequence of T consecutive video frames comprising the video clip. Each frame F_i is considered to be an RGB image of dimensions determined by the resolution of the recording camera. A Multi-Object Tracking (MOT) algorithm applied to $\mathcal{V}$ yields a set of BBs for each frame, along with a track label assigned to each BB. Let $\mathcal{B} = \{B_1, \ldots, B_N\}$ represent the set of all BBs identified by the MOT algorithm. For each $B_i \in \mathcal{B}$, the MOT algorithm provides the following information:

(a) The index of the frame in which the BB appears, denoted t_i,

(b) The coordinates of the BB within the frame, represented as $\langle x_{\text{top}}, y_{\text{top}}, w, h\rangle$,

(c) The assigned track label, denoted y_i^T.

Furthermore, it is assumed that a feature extractor is available for application to the contents of the BB, resulting in:

(d) A feature vector $\mathbf{x}_i$. This vector may be derived from a range of feature extraction methods, such as autoencoders, deep neural networks, Histogram of Oriented Gradients (HOG), or RGB values.

The objective of the task is to assign an identity label to each $B_i \in \mathcal{B}$. This is carried out under the assumption that the approximate number of distinct true identities present in the video is known beforehand. This prior knowledge guides the labelling process by constraining the number of possible identities to be assigned.

To this end, we formulate the animal re-identification problem as a label matching task. Each bounding box B_i is associated with a ground truth label y_i^{GT}. The objective is to infer a predicted label for each B_i by leveraging the track label y_i^T, the feature vector $\mathbf{x}_i$, and the frame number t_i. While conventional classification tasks typically rely solely on the feature representation $\mathbf{x}_i$ to train a classifier, the present setting provides additional contextual information. The proposed method, termed CBC, is specifically designed to integrate all available sources of information in order to estimate y_i^{GT} more accurately.

The CBC method leverages elements (a), (c), and (d). It is assumed that the bounding box coordinates (b) have already been utilised by the MOT algorithm for track construction. To evaluate the utility of post-processing the MOT tracks, we assume access to the ground truth labels, enabling a direct comparison with the output of the CBC method.

Several strategies for addressing this problem are conceivable:

A. A baseline approach involves using the raw track labels (c) as the final identity labels, without applying any further clustering. This baseline is informative, as in scenarios where tracks are pure—that is, each track corresponds to a consistent unique identity—merging may be beneficial. However, in cases where tracks are impure, as is likely with our dataset, such merging may be ineffective or even detrimental.

B. Apply a standard, off-the-shelf clustering algorithm directly to the feature vectors (d), disregarding the track labels entirely. This method relies exclusively on the similarity of the feature representations to group instances, treating each detection independently. While this can be effective when the feature space is well-structured and separable, it neglects potentially valuable temporal and spatial information embedded within the tracking data. Such contextual information may be particularly important in more complex scenarios, such as those involving animal data, where appearance alone may be insufficient for reliable identity resolution.

C. Perform constrained clustering directly on the feature vectors (d), incorporating additional structure through both ML and CL constraints. In this setting, track labels (c) are used to define ML constraints, indicating that all detections within the same track should be assigned to the same cluster. Simultaneously, CL constraints are inferred from frame co-occurrence data (a), under the assumption that two detections appearing in the same frame must belong to different individuals. By integrating both types of constraints, this approach aims to improve clustering performance by leveraging both appearance-based similarity and contextual information derived from temporal and spatial cues.

D. Computing a centroid for each track, denoted μ_i, by averaging the feature vectors (d) associated with that track. These centroids serve as compact representations of the tracks and are subsequently clustered using a conventional clustering algorithm. This method reduces the complexity of the problem by considering a higher-level of abstraction. However, it assumes that the track features are internally consistent and representative of a single identity, which may not hold in the presence of impure tracks.

E. Exploit the implicit ML constraints within tracks and infers CL constraints between tracks using frame index data (a). A constrained clustering algorithm is then applied to the set of track-level centroids, denoted μ_i, as outlined in strategy D. By combining the reduced complexity afforded by track summarisation with spatial and temporal constraints, this approach provides more guidance to the clustering algorithm. However, its performance may still be limited by the presence of impure tracks.

F. Utilise the track labels (c), frame indices (a), and feature vectors (d) jointly, without reducing the data to centroid representations. By preserving the full set of feature vectors, it retains the detailed variability within each track, allowing for a more nuanced analysis of individual detections. This approach aims to exploit all the available information to maximise its performance.

The proposed CBC approach corresponds to Strategy (F). To identify the most effective strategy for the aforementioned task, each strategy is systematically implemented, tested, and compared. For Strategies (B) and (D), we employ standard, readily available clustering algorithms. These include:

1. k-means
2. Single Linkage: The merging criterion employed in this approach is defined by the minimum distance between any single pair of elements belonging to two distinct clusters.
3. Average Linkage: The merging criterion in this approach is defined by the average distance computed over all pairs of points, with each pair comprising one point from each of the two clusters under consideration.
4. Centroid Linkage: The merging criterion in this approach is defined by the distance between the centroids (or centres) of the two distinct clusters under consideration.
5. Complete Linkage: The merging criterion in this approach is defined by the maximum distance between any pair of elements belonging to two distinct clusters.
6. Median Linkage: The merging criterion is determined by the distance between the centroids (or centres) of the two distinct clusters under consideration. Following the merging of two clusters, the centroid of the resultant cluster is calculated as the midpoint of the centroids of the merged clusters.
7. Ward Linkage: The merging criterion in Ward's method is defined by the increase in the total within-cluster variance (or error sum of squares) that would result from merging two clusters. At each step, the pair of clusters whose merger results in the minimum increase in this variance is chosen.
8. Weighted Linkage: The merging criterion is based on the average distance between clusters. Following a merger, the distance between the newly formed cluster and any other cluster is computed as the average of the distances from each original cluster to the remaining cluster, assigning equal weight to both.

9. FINCH. First Integer Neighbour Clustering Hierarchy [148] is a parameter-free, hierarchical clustering algorithm designed for efficient and scalable clustering of large datasets. Instead of relying on pairwise distance thresholds, FINCH builds clusters by connecting each data point to its first nearest neighbour based on similarity, forming initial groups that are then recursively merged to produce a clustering hierarchy. This approach reduces computational complexity and automatically determines the number of clusters at different levels, making it particularly useful in applications such as person re-identification [34]. Our implementation utilises the default parameters as provided in the MATLAB code released by the original authors.

10. Gaussian Mixture Models (GMMs). These probabilistic models are employed for clustering and density estimation, operating under the assumption that data points are generated from a mixture of a finite number of Gaussian distributions, each characterised by its own mean and covariance matrix. Gaussian Mixture Models (GMMs) are particularly effective for identifying underlying groups (clusters) within datasets, especially when these groups overlap or exhibit complex shapes. In our implementation, we utilised diagonal shared covariance matrices and set the maximum number of iterations to 1000.

11. DBSCAN. DBSCAN [65] is a density-based clustering algorithm that groups together points in high-density regions while designating points in low-density areas as outliers (noise). It relies on two parameters: ϵ, the radius within which neighbouring points are considered, and $MinPts$, the minimum number of points required to form a dense region. We observed that the choice of the maximum distance parameter ϵ significantly affects the clustering results. Consequently, we ran DBSCAN over a range of ϵ values, specifically $\epsilon \in \{0.5, 1.0, \ldots, 4.0\}$. For each ϵ, we varied $MinPts$ across $\{1, 2, \ldots, 8\}$. Finally, we selected the combination of ϵ and $MinPts$ that produced a number of clusters closest to the desired count.

12. Spectral Clustering. Spectral Clustering [133] is a technique that utilises the eigenvalues and eigenvectors of a similarity matrix derived from the data to perform clustering. Instead of clustering directly in the original feature space,

spectral clustering projects the data into a lower-dimensional space by leveraging the spectrum (eigenvectors) of a graph Laplacian constructed from pairwise similarities between data points. This approach is particularly effective for identifying clusters that are neither necessarily spherical nor linearly separable, allowing it to capture complex cluster structures based on the connectivity among data points.

For approaches C and E, we use a Constrained Clustering Ensemble method (CCEN) detailed in Section 6.2, which proved to be the best option for our type of data. CCEN incorporates temporal pairwise ML and CL constraints. We found the optimum parameters for this method to be an ensemble size of 5, using average linkage as the base clusterer.

Proposed Method

The proposed CBC method seeks to utilise the track labels produced by a MOT algorithm without collapsing—and thereby oversimplifying—the tracks into their corresponding centroid descriptors. Furthermore, by combining frame indices with track labels, instance-level CL constraints can be constructed and subsequently elevated to track-level CL constraints. These constraints indicate whether two tracks should not be linked, based on the fact that elements of each appear in the same frame at some point in the video.

The method begins by training a chosen classifier f using the feature descriptors $\mathbf{x}_i$ of the BBs and the track labels $\mathcal{Y}^T$ produced by the MOT algorithm. Once the classifier has been trained, the feature descriptors are resubstituted into f to generate a new set of predicted track labels. A confusion matrix $\mathbf{M}$ is subsequently computed between the original track labels and the predicted track labels.

Before the confusion matrix $\mathbf{M}$ can be used to merge tracks, it is first necessary to set its leading diagonal to zero; that is, $M_{ij} = 0$ where $i = j$, since correctly assigned data should not influence the decision to merge two distinct tracks. Next, the track-level CL constraints must be incorporated into the confusion matrix. Let y and z denote two distinct tracks (classes) such that $(y, z) \in CL$; then we set $M_{y,z} = M_{z,y} = 0$. This ensures that no two tracks between which a CL constraint exists can be merged.

The remaining non-zero values of the confusion matrix, i.e. where $M_{ij} > 0$, indicate the degree of similarity between the feature representations of two distinct tracks. Since the classifier was trained using the feature vectors $\mathbf{x}_i$, its predictions reflect similarity in appearance. Let p and q denote two different tracks (classes) such that $M_{p,q} > 0$. A higher value of $M_{p,q}$ implies that the trained classifier f has more frequently misclassified objects from track p as belonging to track q. This suggests that p and q are similar in the feature space and may, in fact, correspond to the same underlying identity. However, rather than relying on the absolute number of misclassifications, we are interested in the proportion of track p that has been labelled as q. To this end, each row of $\mathbf{M}$ is normalised such that the sum of its entries equals one.

The largest entry of the confusion matrix $\mathbf{M}$, denoted $M_{i,j}$, is then used to merge tracks i and j into a single track, and the track labels $\mathcal{Y}^T$ are updated accordingly. This process is repeated iteratively until one of two conditions is met: either all values of $M_{i,j} = 0$, indicating that no further merges can be performed without violating CL constraints, or the predefined number of clusters K has been reached as a result of previous merges.

The proposed algorithm is ’monolithic’ in nature, as it is grounded in a single core principle: constrained clustering through classification. Among its components, the only aspect that appears suitable for removal in the context of an ablation study is the scaling of the confusion matrix $\mathbf{M}$. This was explored in a preliminary experiment; however, the results deteriorated significantly in the absence of this scaling step.

The algorithm is detailed in Algorithm 3:

6.1.2 Experimental Study

Data

The datasets employed in this study comprised the five video sequences described in Chapter 3, supplemented by ten additional videos sourced from the Edinburgh Pig Behaviour Video Dataset [23], the characteristics of which are summarised in Table 6.1.

The Edinburgh Pig dataset (EP) consists of ten videos depicting eight pigs within a single enclosure, recorded using a stationary, top-down camera positioned above the pen. As with the five videos discussed in Chapter 3, the scenes are densely populated,

Algorithm 3 Classifier-Based Clustering (CBC)

```
Input: Dataset $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, frame indices $\mathcal{T} = \{t_1, \dots, t_N\}$, track labels
    $\mathcal{Y}^T = \{y^t_1, \dots, y^t_N\}$, desired number of clusters $K$ (assume $K$ is smaller than
    number of tracks).
Output: $\mathcal{Y}$
Select classifier model $f$. Set current label set $\mathcal{Y} \leftarrow \mathcal{Y}^T$.
Initialise: True do
    Train classifier $f$ on dataset $\mathcal{X}$ with labels $\mathcal{Y}$.
    Relabel all data in $\mathcal{X}$ using $f$ (resubstitution); construct confusion matrix $\mathbf{M}$.
    Set the diagonal of $\mathbf{M}$ to zero.
    Identify CL constraints from $\mathcal{T}$ and $\mathcal{Y}$. For each $(p, q) \in CL$, set $M_{pq} =$
    $M_{qp} = 0$.
    Row-normalise $\mathbf{M}$ so that each row sums to 1.
    Identify the largest entry $M_{ij}$.
    if $M_{ij} = 0$ or number of unique labels in $\mathcal{Y}$ equals $K$ then
        return current label set $\mathcal{Y}$.
    else
        Merge tracks $i$ and $j$: relabel all points in $\mathcal{Y}$ with label $i$ to label $j$.
    end if
end Initialise:
```

leading to frequent instances of occlusion between animals. Nonetheless, there are several key differences between the EP videos and those described in Chapter 3. Firstly, because the pigs are confined to the enclosure, individuals cannot leave or re-enter the frame, ensuring that the number of animals remains constant throughout each video. Secondly, the EP data have been annotated only at selected intervals, with occasional omissions of certain animals, in contrast to the continuous annotations provided in the benchmark videos. This sparser annotation presents challenges for machine learning applications, as the displacement of animals between consecutive annotated frames may be too great to support the reliable construction of robust constraints.

Although the Edinburgh Pig videos do not encapsulate all the challenging conditions inherent in the bespoke benchmark datasets introduced in Chapter 3, they nonetheless offer valuable insights into the performance of the proposed method (CBC). Throughout the experiments, RGB-based feature representations were utilised, as outlined in Section 3.1.

Track Generation

A typical tracking algorithm comprises two primary stages: object detection and association [164, 193]. In each video frame, the detection stage identifies a set of BBs,

Table 6.1: Characteristics of the ten Edinburgh Pigs videos videos

Video	T	L	N	c	Min p/f	Max p/f	Avr p/f	Imbalance
EP000002	600	60	3586	8	2	8	6.0	1.7
EP000005	185	60	231	8	1	8	1.2	26.5
EP000009	310	60	1361	8	1	8	4.4	1.6
EP000010	480	60	913	8	1	8	1.9	3.5
EP000016	375	60	577	8	1	8	1.5	3.0
EP000028	312	60	440	8	1	8	1.4	11.0
EP000033	483	60	979	8	1	8	2.0	1.6
EP000036	414	60	699	8	1	8	1.7	10.6
EP000060	169	60	198	8	1	8	1.2	39.2
EP000078	280	60	373	8	1	8	1.3	9.9

Table notes: T is the number of frames; L is the video length in seconds; N is the number of objects (individual animal clips); c is the number of classes (animal identities); Min p/f is the minimum number of animals per frame (image); Max p/f and Avr p/f are respectively the maximum and the average numbers; Imbalance represents the size of the largest class divided by the size of the smallest class.

and the association stage links these BBs across consecutive frames to form partial segments of the trajectory corresponding to the tracked object known as tracklets. In this study, we chose to bypass the detection stage by supplying the tracking algorithms with ground truth BBs—those derived from manual video annotation. As a result, the tracking algorithms executed only the association stage. This approach effectively eliminates potential errors stemming from imperfect object detection, thereby allowing us to evaluate how well tracking can support animal re-identification under the most favourable conditions.

For the purposes of our experiments, we generated three sets of track files for each video:

- *MATLAB:* The first tracking method employed the standard MATLAB tracking algorithm available through the Automated Driving Toolbox. In this approach, ground truth BBs were directly inserted into the algorithm in place of automatically detected ones. The tracking process relies on a multi-object tracker that predicts the positions of BBs in subsequent frames using a Kalman filter and performs association via the Global Nearest Neighbour algorithm[1]. Importantly, BB

[1] https://uk.mathworks.com/help/driving/ref/multiobjecttracker-system-object.html

appearance features are not used during track formation. Track collisions are resolved based on predicted motion trajectories.

- *BASIC:* The second method is based on temporal ML constraints, whereby BBs in adjacent frames are linked if their intersection over union (IoU) exceeds a specified threshold. In our implementation, this threshold was set to 0.7, corresponding to a 70% overlap. For each pair of consecutive frames, the algorithm computes the IoU for all possible BB pairs (one from each frame), and the Munkres algorithm (Hungarian method) is used to determine the optimal one-to-one assignments. This approach is particularly effective in scenarios involving occlusion, where multiple objects may compete to match with a single candidate BB in the adjacent frame.

- *FCG:* The third approach leverages Feature Combinatorial Grouping (FCG) [74], which is predicated on the assumption that instances of the same object will exhibit similar visual features over short temporal intervals. FCG operates in two stages. Initially, it constructs a set of short tracklets. In the subsequent stage, these tracklets are merged hierarchically over time through a process informed by so-called "lifted frames"—aggregated intervals that group tracklets rather than individual detections. Clustering is performed using the UPGMA (Unweighted Pair Group Method with Arithmetic Mean) algorithm, which iteratively merges the most similar cluster pairs, yielding a hierarchical structure from which the final tracks are derived. In our implementation, we integrated RGB feature representations into the FCG pipeline to ensure a fair comparison with the other methods.

Experimental Protocol

The objective of this experiment is to evaluate the six strategies outlined in Section 6.1.1 for the task of animal re-identification. Accordingly, the experiment is organised into six corresponding sections, each associated with one of the strategies, labelled A through F.

Approach A involves directly comparing the track labels generated by each of the three tracking methods — MATLAB, BASIC, and FCG — to the ground truth labels. This yields three evaluation results per dataset and serves as a baseline for assessing the reliability of track labels without any additional clustering or refinement.

Approach B applies twelve off-the-shelf clustering algorithms to the raw feature data, as enumerated in Section 6.1.1. Each algorithm is independently applied to the dataset, resulting in twelve clustering outcomes for each video sequence.

In Approach C, constrained clustering is applied to the raw data using both ML and CL constraints. The ML constraints are derived from the tracks produced by each of the three tracking methods, while the CL constraints are inferred based on the co-occurrence of objects within the same video frame. The clustering algorithm employed in this approach is CCEN, resulting in three outcomes — one for each set of ML constraints.

Approach D focuses on clustering the centroids of the tracks, which are computed from the bounding box (BB) features. These centroids form a new dataset, to which the same twelve clustering algorithms are applied. Given the three sources of track data (*MATLAB*, *BASIC*, and *FCG*), this yields a total of $12 \times 3 = 36$ experimental results.

In Approach E, the CCEN clustering algorithm is again employed, this time on the dataset composed of track centroids. One result is produced for each of the three sets of tracks, resulting in three evaluations.

Finally, Approach F involves the application of the proposed Classifier-Based Clustering (CBC) method. CBC is executed on the raw data using the initial class labels provided by each of the three tracking methods. This configuration produces three additional results.

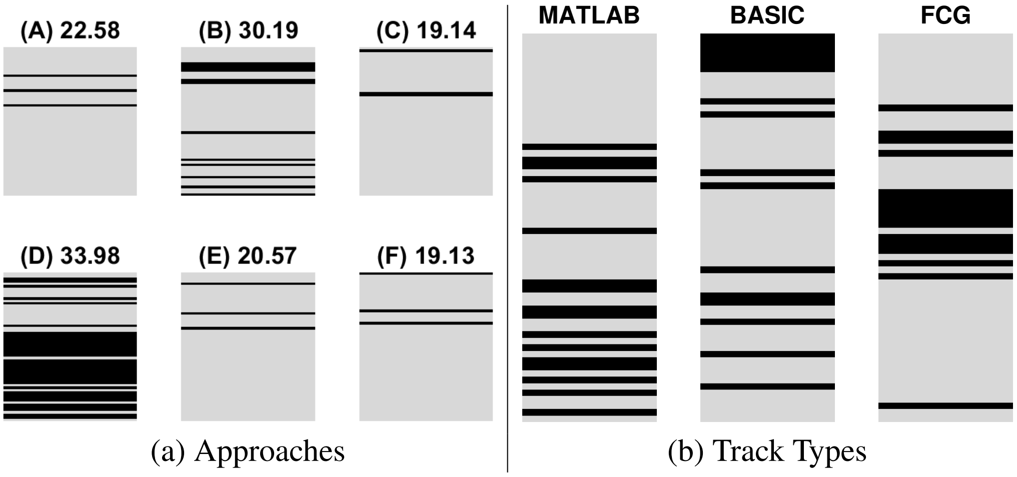
In total, each dataset yields 60 experimental outcomes. For every configuration, the assigned cluster labels are evaluated against the ground truth labels using the ARI, providing a quantitative measure of clustering accuracy.

6.1.3 Results

Table 6.2 presents the ARI values obtained in the experiment. Based on these results, we prepared Table 6.3, where the 60 methods are ranked from best (lowest rank) to worst. The best-performing method is our proposed CBC with Basic Tracks. However, no single method is universally superior across all datasets.

To determine which group of methods is significantly better than the others, we apply the Friedman test incrementally. Initially, we compare the top two methods; subsequently, we add one method at a time, calculating the p-value for the hypothesis that the methods within the group are indistinguishable. A cut-off of $p < 0.05$ is used to identify the largest group of methods at the top of the ranking that cannot be statistically distinguished. This threshold is indicated by a horizontal line in Table 6.3.

Figure 6.1 provides a visual representation of the performance of the sixty methods across our datasets. The rankings from Table 6.3 are illustrated as a grey block, with black stripes marking the presence of keywords in the method labels. Subplot (a) depicts the positions of the six approaches, with the average rank for each approach shown above the respective block (lower values indicate better performance). Subplot (b) presents the results grouped by track type.



(a) Approaches (b) Track Types

Figure 6.1: Position of the category in the ranking table. The higher the position, the better the category against the alternative. The average rank for the category is also shown. [188]

Based on the results presented in Table 6.3 and Figure 6.1, several observations can be made. The proposed CBC method (F) achieves the best overall performance when applied with the BASIC tracks. However, the advantage of CBC is only marginal, with constrained clustering of the raw data (C) closely following in terms of effectiveness.

Despite this, there is no definitive winner among the methods or approaches tested. While CBC (F) holds a slightly better overall rank than constrained clustering of the raw data (C), the difference is minimal, indicating that multiple approaches perform comparably on the datasets examined.

Interestingly, non-constrained clustering of track centroids (D) performs worse than the direct use of track labels alone (A). This finding challenges the commonly held assumption in current research that clustering track centroids generally improves results. We attribute this outcome to the intrinsic structure of our datasets, which limits the applicability of more advanced feature extraction techniques, such as deep features.

Furthermore, constrained clustering approaches (F), (C), and (E) consistently outperform the tracks-only baseline (A), supporting our primary assertion. Consequently, we recommend the use of constrained post-clustering of tracks as a promising direction for future work.

Finally, it appears that sophisticated tracking methods may not necessarily yield better results on this type of data. The simple BASIC tracks method, based on intersection over union (IoU) matching, proves sufficiently effective in this context.

Table 6.2: Percentage ARI scores for all methods as they perform on the video datasets.

	EP000002	EP000005	EP000009	EP000010	EP000016	EP000028	EP000033	EP000036	EP000060	EP000078	Koi Fish	Pigeons (Ground)	Pigeons (Kerb)	Pigeons (Square)	Pigs
(A) Tracks-Only MATLAB	10	4	6	6	7	7	5	6	17	6	19	17	5	15	2
(A) Tracks-Only BASIC	5	5	4	3	2	3	2	2	13	2	64	54	9	26	3
(A) Tracks-Only FCG	1	34	1	9	15	22	4	12	91	27	13	5	1	1	1
(B) Raw - Single linkage	0	4	0	0	15	0	0	0	100	0	0	0	0	0	0
(B) Raw - Complete linkage	2	17	2	4	23	9	2	9	49	17	15	9	10	24	10
(B) Raw - Average linkage	0	19	1	0	13	8	0	8	75	16	8	5	1	12	3
(B) Raw - Weighted linkage	1	19	4	3	24	7	1	6	75	10	16	11	6	18	10
(B) Raw - Centroid linkage	0	4	0	0	6	2	0	2	75	0	4	1	1	0	1
(B) Raw - Median linkage	0	6	0	0	19	3	0	1	75	3	1	1	1	4	0
(B) Raw - Ward linkage	4	18	4	5	30	12	4	20	24	18	15	13	16	36	18
(B) Raw - Kmeans	3	20	3	4	28	13	4	14	48	15	18	14	19	35	16
(B) Raw - GMM	3	22	4	6	29	14	5	14	22	17	22	10	16	32	16
(B) Raw - FINCH	3	19	3	3	25	16	3	15	27	17	19	13	18	34	19

Continued on next page

Table 6.2 – continued from previous page

	EP000002	EP000005	EP000009	EP000010	EP000016	EP000028	EP000033	EP000036	EP000060	EP000078	Koi Fish	Pigeons (Ground)	Pigeons (Kerb)	Pigeons (Square)	Pigs
(B) Raw - Spectral	0	7	0	1	7	4	0	13	62	9	8	0	0	0	3
(B) Raw - DBSCAN	0	4	0	1	3	0	0	1	-7	3	0	0	0	0	0
(C) Raw - CCEN MATLAB	5	7	4	4	8	7	4	7	84	5	15	11	5	14	2
(C) Raw - CCEN BASIC	2	23	2	5	65	16	6	21	49	24	52	43	8	25	2
(C) Raw - CCEN FCG	0	34	1	8	15	22	4	11	91	27	13	5	1	1	1
(D) MATLAB - Single linkage	0	0	0	0	0	0	0	0	70	0	4	1	0	5	1
(D) MATLAB - Complete linkage	2	5	0	0	1	2	1	-1	35	5	10	3	3	8	2
(D) MATLAB - Average linkage	0	3	0	0	-1	1	0	1	85	0	6	3	0	6	2
(D) MATLAB - Weighted linkage	0	7	0	1	-1	4	0	1	85	3	6	3	1	6	2
(D) MATLAB - Centroid linkage	0	1	0	0	-1	1	0	0	85	0	5	1	0	5	1
(D) MATLAB - Median linkage	0	2	0	0	-1	1	0	1	85	0	5	1	0	5	1
(D) MATLAB - Ward linkage	2	10	0	1	1	6	1	1	35	5	6	11	3	9	2
(D) MATLAB - Kmeans	1	9	0	0	5	7	1	1	85	4	6	4	3	9	2
(D) MATLAB - GMM	1	2	0	1	3	2	0	2	0	5	0	9	2	0	0
(D) MATLAB - FINCH	2	0	8	4	6	7	4	1	31	4	15	17	4	10	1
(D) MATLAB - Spectral	1	1	0	0	0	0	0	0	85	0	6	12	3	13	2
(D) MATLAB - DBSCAN	0	5	0	0	4	4	0	1	28	1	5	1	0	6	0
(D) BASIC - Single linkage	0	4	0	0	4	0	0	0	100	1	0	3	0	0	0
(D) BASIC - Complete linkage	3	17	4	4	24	11	3	14	100	16	23	23	0	11	1
(D) BASIC - Average linkage	1	17	0	0	12	5	0	2	100	17	8	7	0	3	0
(D) BASIC - Weighted linkage	2	19	2	3	12	10	0	16	100	7	23	15	1	6	0
(D) BASIC - Centroid linkage	0	3	0	0	8	0	0	3	100	0	0	6	0	1	0
(D) BASIC - Median linkage	0	7	0	0	11	3	0	3	100	0	7	8	0	1	0
(D) BASIC - Ward linkage	5	16	4	4	25	12	8	20	100	23	22	23	1	12	2
(D) BASIC - Kmeans	5	18	4	5	37	12	7	17	100	14	42	28	4	13	2
(D) BASIC - GMM	4	17	4	6	34	13	5	18	0	20	27	29	1	14	2
(D) BASIC - FINCH	5	19	4	5	26	10	6	18	46	15	47	45	9	21	4
(D) BASIC - Spectral	0	15	0	1	28	9	0	11	100	14	0	27	0	5	1
(D) BASIC - DBSCAN	2	5	0	1	3	0	0	3	38	7	19	10	1	3	0
(D) FCG - Single linkage	0	34	0	0	15	21	1	6	91	23	13	0	1	1	1
(D) FCG - Complete linkage	0	34	1	2	15	21	2	6	91	23	13	2	1	1	1
(D) FCG - Average linkage	0	34	1	0	15	21	1	6	91	23	13	2	1	1	1
(D) FCG - Weighted linkage	0	34	1	0	15	21	1	6	91	23	13	2	1	1	1
(D) FCG - Centroid linkage	0	34	0	0	15	21	1	6	91	23	13	1	1	1	1

Continued on next page

Table 6.2 – continued from previous page

	EP000002	EP000005	EP000009	EP000010	EP000016	EP000028	EP000033	EP000036	EP000060	EP000078	Koi Fish	Pigeons (Ground)	Pigeons (Kerb)	Pigeons (Square)	Pigs
(D) FCG - Median linkage	0	34	0	0	15	21	1	6	91	23	13	1	1	1	1
(D) FCG - Ward linkage	0	34	1	2	15	21	2	6	91	23	13	2	1	1	1
(D) FCG - Kmeans	0	31	1	4	15	20	1	6	91	23	13	1	1	1	1
(D) FCG - GMM	0	0	0	0	15	0	0	0	91	0	13	0	0	0	0
(D) FCG - FINCH	0	22	1	4	15	9	1	-1	91	10	13	3	0	0	1
(D) FCG - Spectral	0	32	1	3	15	21	2	6	91	23	13	4	1	1	1
(D) FCG - DBSCAN	0	33	0	0	15	21	1	6	91	23	13	0	1	1	1
(E) CCEN MATLAB	6	5	5	2	10	5	5	4	84	3	15	10	4	14	2
(E) CCEN BASIC	2	18	2	4	67	15	5	19	49	23	60	45	7	25	2
(E) CCEN FCG	0	34	1	8	15	22	3	11	91	27	13	5	1	1	1
(F) CBC MATLAB	10	6	5	3	11	4	5	7	79	6	12	12	5	14	2
(F) CBC BASIC	10	13	6	6	36	9	6	13	94	11	74	51	9	27	2
(F) CBC FCG	1	33	1	8	15	22	4	12	91	24	13	5	1	1	1

Table 6.3: Friedman test on all methods from our proposed approaches.

Method	ARI	Rank	p-value
(F) CBC BASIC	0.2449	10.8000	-
(C) Raw - CCEN BASIC	0.2286	11.9667	0.1967
(D) BASIC - Kmeans	0.2049	12.3667	0.5488
(D) BASIC - FINCH	0.1864	12.8667	0.5641
(E) CCEN BASIC	0.2283	14.1000	0.4873
(D) BASIC - Ward linkage	0.1838	14.1667	0.4159
(B) Raw - Ward linkage	0.1586	14.2667	0.6483
(B) Raw - GMM	0.1545	14.4667	0.8166
(B) Raw - Kmeans	0.1699	14.8000	0.8815
(B) Raw - FINCH	0.1563	15.6667	0.8991
(D) BASIC - GMM	0.1283	16.8333	0.7936
(A) Tracks-Only FCG	0.1575	19.9333	0.5244

Continued on next page

Table 6.3 – continued from previous page

Method	ARI	Rank	p-value
(D) BASIC - Complete linkage	0.1686	20.1667	0.1328
(B) Raw - Complete linkage	0.1346	20.3333	0.0169
(B) Raw - Weighted linkage	0.1406	21.9000	0.0012
(F) CBC FCG	0.1538	22.3333	0.0011
(E) CCEN FCG	0.1551	22.5000	0.0011
(A) Tracks-Only MATLAB	0.0870	22.6667	0.0010
(C) Raw - CCEN FCG	0.1552	22.7000	0.0010
(C) Raw - CCEN MATLAB	0.1209	22.7667	0.0003
(F) CBC MATLAB	0.1194	24.2667	0.0001
(D) BASIC - Weighted linkage	0.1439	25.1000	0
(E) CCEN MATLAB	0.1158	25.1000	0
(A) Tracks-Only BASIC	0.1313	25.1333	0
(D) FCG - Complete linkage	0.1416	27.4333	0
(D) FCG - Ward linkage	0.1418	27.5000	0
(D) FCG - Spectral	0.1415	27.9000	0
(D) FCG - Weighted linkage	0.1399	27.9333	0
(D) FCG - Kmeans	0.1383	28.2333	0
(D) FCG - Average linkage	0.1397	28.6667	0
(D) MATLAB - FINCH	0.0751	29.0000	0
(D) FCG - DBSCAN	0.1369	30.1333	0
(D) FCG - Centroid linkage	0.1386	30.6333	0
(D) FCG - Median linkage	0.1386	30.7667	0
(B) Raw - Average linkage	0.1122	31.1667	0
(D) FCG - Single linkage	0.1380	31.3333	0
(D) BASIC - Spectral	0.1410	32.3667	0
(D) FCG - FINCH	0.1132	33.8000	0
(D) MATLAB - Ward linkage	0.0619	34.8333	0
(D) MATLAB - Kmeans	0.0914	35.7333	0
(D) BASIC - Average linkage	0.1140	37.3667	0
(D) BASIC - DBSCAN	0.0612	37.8667	0

Continued on next page

Table 6.3 – continued from previous page

Method	ARI	Rank	p-value
(D) MATLAB - Complete linkage	0.0505	39.5000	0
(D) MATLAB - Spectral	0.0821	40.1333	0
(D) BASIC - Median linkage	0.0940	40.1667	0
(B) Raw - Spectral	0.0767	40.2000	0
(D) MATLAB - Weighted linkage	0.0785	41.2667	0
(B) Raw - Median linkage	0.0770	41.3000	0
(D) MATLAB - GMM	0.0185	44.0667	0
(D) BASIC - Centroid linkage	0.0806	46.1000	0
(D) MATLAB - Average linkage	0.0705	46.2667	0
(D) MATLAB - DBSCAN	0.0368	46.7667	0
(B) Raw - Centroid linkage	0.0633	47.3000	0
(D) MATLAB - Centroid linkage	0.0666	48.3667	0
(D) BASIC - Single linkage	0.0748	48.4333	0
(D) MATLAB - Median linkage	0.0670	48.5000	0
(B) Raw - Single linkage	0.0798	48.5667	0
(D) FCG - GMM	0.0789	49.6333	0
(D) MATLAB - Single linkage	0.0543	51.2333	0
(B) Raw - DBSCAN	0.0037	52.3333	0

6.2 Constrained Clustering Ensemble ($CCEN$)

When monitoring animals in video footage, data are expected to evolve gradually over time. Although predicting the overall progression of data across an entire video sequence is challenging, it is reasonable to assume that, over shorter intervals—such as between consecutive frames—the appearance of an object changes only marginally. Consequently, its feature representation also exhibits minimal variation, regardless of the specific feature extraction technique employed. From this premise, it can be inferred that data points within a video sequence tend to form string-like clusters in feature space.

However, as the same object may reappear at different times, multiple string-shaped clusters can emerge, potentially occupying distinct regions of the feature space.

The results of applying constrained clustering to complete animal video datasets highlight the limitations of such methods in effectively addressing re-identification tasks. This is evident from the performance of the methods compared in Section 6.1, as well as from the observed decline in performance with increasing window size, as demonstrated in Section 5.1. These findings suggest that, even in offline contexts, reducing structural complexity through a windowed processing approach may provide a more practical solution for offline animal re-identification, as it facilitates the identification of string-shaped clusters within smaller segments of the dataset.

Although several clustering strategies may be applicable to this problem, recent advances in constrained clustering have increasingly prioritised accuracy, resulting in highly complex algorithms. This complexity often comes at the expense of computational efficiency and interpretability, rendering such methods less suitable for online learning or real-time applications.

While a single clustering method may suffice in certain scenarios, extensive research in the field has demonstrated that cluster ensembles generally outperform individual clustering algorithms [31, 75, 115]. Accordingly, this work proposes the development of a clustering ensemble composed of simple, semi-supervised agglomerative hierarchical methods, denoted as $CCEN$.

6.2.1 Proposed Method

To meet the requirement for time-efficient and reliable hierarchical clustering approaches, we adopt the agglomerative hierarchical constrained clustering method proposed by Klein et al., as described in Section 5.1.3, as the foundational technique for our ensemble, denoted as $CCEN$.

The algorithm accepts as input the dataset $\mathcal{X}$ containing N objects, the sets of instance-level constraints $\mathcal{ML}$ and $\mathcal{CL}$, the chosen constrained clustering base method $CCBM$, the desired number of clusters K, and the number of ensemble members E. The $CCEN$ algorithm begins by initialising an empty cumulative adjacency matrix $\mathbf{M}^{CA}$ of size

$N \times N$. Each ensemble member contributes an individual adjacency matrix, which is then added to $\mathbf{M}^{CA}$. As a result, pairs of points that are consistently grouped together across all ensemble members will have an entry of E in $\mathbf{M}^{CA}$, while pairs never clustered together will have an entry of zero.

By employing a single base hierarchical clustering method across all ensemble constituents, we execute the base method only once and reuse the resulting dendrogram. This dendrogram is then cut at successive levels to produce clusterings with $K, K+1, \ldots, K+E$ clusters. In this way, we construct the ensemble without incurring the computational cost of running the clustering algorithm multiple times, thereby preserving the efficiency of a single-method approach.

Once $\mathbf{M}^{CA}$ has been completed, it is converted into cluster labels. This is achieved by first transforming $\mathbf{M}^{CA}$ into a binary adjacency matrix through thresholding: any element in $\mathbf{M}^{CA}$ greater than $\frac{E}{2}$ is set to 1, while those less than or equal to $\frac{E}{2}$ are set to 0. The resulting ensemble adjacency matrix, denoted $\mathbf{M}^{EA}$, defines an undirected graph, whose connected components represent the final clustering structure. These components are returned by the $CCEN$ algorithm as the ensemble cluster labels $\mathcal{Y}^{E}$. The full procedure is summarised in Algorithm 4.

6.2.2 Experimental Study

Data

The datasets used in this experiment differ from those described in Chapter 3. Instead, a contributed collection of synthetic datasets was utilised, encompassing a wide variety of cluster shapes and varying degrees of clustering difficulty. All datasets in this collection have previously been employed in published studies on clustering. The synthetic collection includes 47 two-dimensional (2D) datasets and 5 three-dimensional (3D) datasets. A 2D visualisation of each synthetic dataset is shown in Figure 6.2, while Table 6.4 provides details on the number of objects (N), number of classes (C), and number of features (d), along with corresponding identifiers for cross-referencing with the figure.

Algorithm 4 Constrained Cluster Ensemble (CCEN)

```
Input: Dataset $X$, Must-Link Constraints $ML$, Cannot-Link constraints $CL$ , Constrained Clustering base method $CCBM$, Number of Clusters $K$, Number of Ensemble members $E$
Output: Ensemble labels $\mathcal{Y}^E$
Initialise the Cumulative Adjacency Matrix $\mathbf{M}^{CA} = [\ ]$
for $i$ in $1, \ldots, E$ do
    $L \leftarrow$ Calculate the cluster labels using base clustering method $CCBM(X, ML, CL, k + i - 1)$
    $\mathbf{M}^{AD} \leftarrow$ calculate the adjacency matrix from returned labels $L$
    $\mathbf{M}^{CA} \leftarrow$ Update the cumulative adjacency matrix by adding the adjacency matrix $\mathbf{M}^{CA} + \mathbf{M}^{AD}$
end for
Initialise the ensemble adjacency matrix $\mathbf{M}^{EA}$
if the value in the cumulative adjacency matrix is greater than half the number of ensemble members $\mathbf{M}^{CA}_{ij} > \frac{E}{2}$ then
    Set the value of the corresponding position in the ensemble adjacency matrix to one $\mathbf{M}^{EA}_{ij} = 1$
else
    Set the value of the corresponding position in the ensemble adjacency matrix to zero $\mathbf{M}^{EA}_{ij} = 0$
end if
$\mathcal{Y}^E \leftarrow$ convert the ensemble adjacency matrix to cluster labels by extracting connected components from the corresponding undirected graph Convert($\mathbf{M}^{EA}$)
```

In addition to the synthetic datasets, a total of 95 widely used real-world datasets were sourced from the UCI Machine Learning Repository [99]. Detailed information about these datasets can be found in Table 6.5.

Methods

The methods we chose for the experiments are:

- COP-KMeans (COP) [173] — This algorithm assigns data points in a manner similar to its unsupervised counterpart, K-Means. However, during the assignment step, a data point is only allocated to a cluster if doing so does not violate any of the specified constraints. If a constraint is breached, the algorithm attempts to assign the point to an alternative valid cluster; if no such assignment is possible, it may terminate with failure. This method will serve as a baseline in the experimental evaluation.

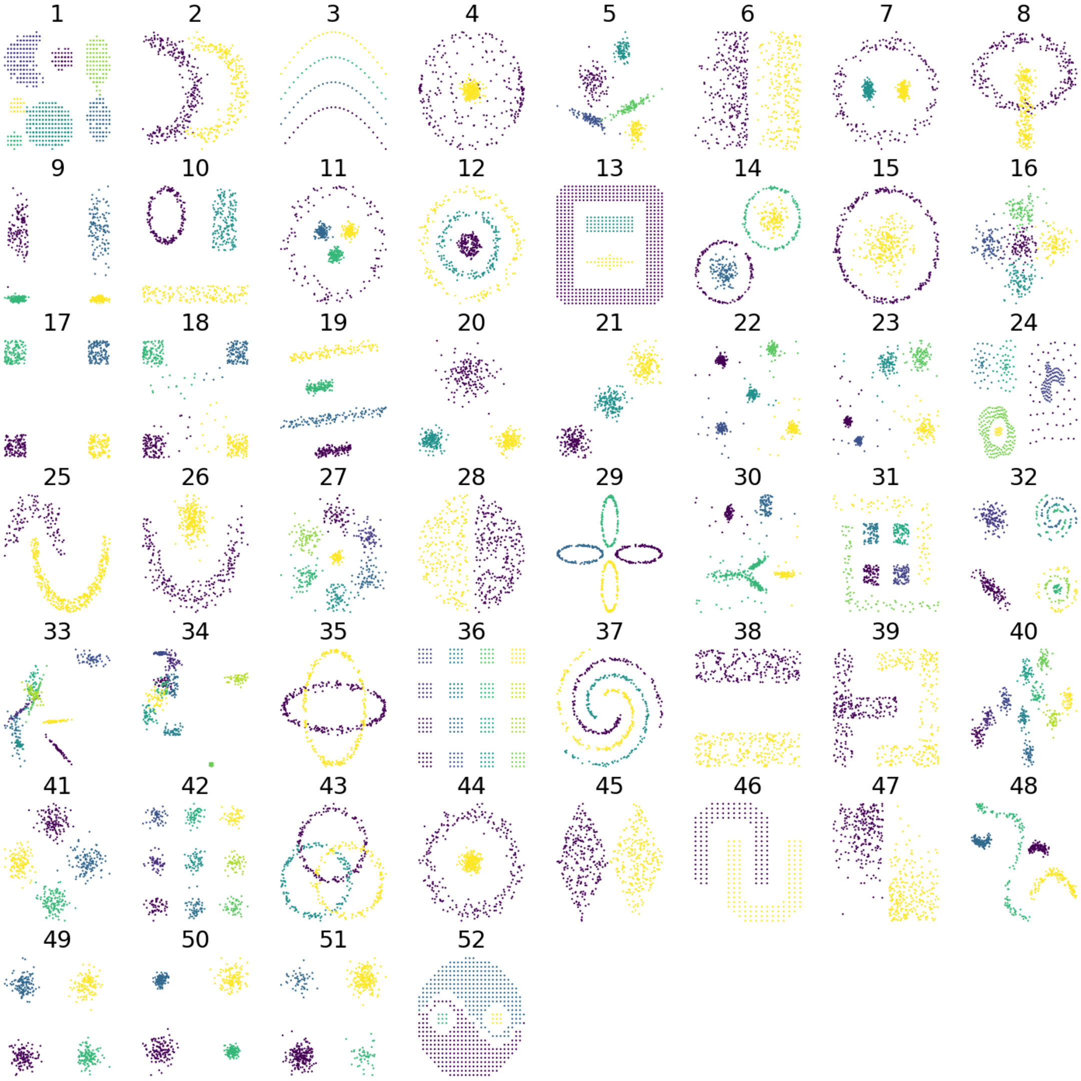


Figure 6.2: Visualises all synthetic datasets employed in the experimental study within a two-dimensional space, allowing for an intuitive understanding of the underlying cluster structures. Each cluster within a dataset is represented by a unique colour, facilitating clear distinction and comparison of the different groupings across the datasets.[189]

Table 6.4: Details of the Synthetic datasets.

#	Dataset	N	C	d
1	Aggregation	788	7	2
2	Aligned_bananas	500	2	2
3	Arcs	104	4	2
4	Atom	500	2	3
5	Balls_and_baguettes	500	5	2
6	Bars	500	2	2
7	Boat	500	3	2
8	Chainlink	500	2	3
9	Cigar	500	4	2
10	Circle_2_rectangles	500	3	2
11	Circle_and_3_gaussians	500	4	2
12	Concentric_circles_3	500	3	2
13	Enclosure	622	3	2
14	Filled_circle_2	500	2	2
15	Filled_circle	500	4	2
16	Flower	500	5	2
17	Four_corners_clear	500	4	2
18	Four_corners_noise	500	4	2
19	Four_lines	500	4	2
20	Gaussians_1_big_2_small	500	3	2
21	Gaussians_3_touching	500	3	2
22	Gaussians_5_compact	500	5	2
23	Gaussians_5_unequal	500	5	2
24	Gestalt	399	6	2
25	Half_rings	500	2	2
26	Happy_wave	500	2	2
27	Hepta	500	7	3
28	Orange	500	2	2
29	Petals	500	4	2
30	Random1	500	4	2
31	Random2	500	6	2
32	Random3	500	7	2
33	Randomised_normal	500	9	2
34	Randomised_triangle	500	10	2
35	Saturn	500	2	2
36	Sixteen_blocks	256	16	2
37	Spirals	500	3	2
38	Stormclouds	500	2	2
39	T_and_u	500	2	2
40	Ten_spherical	500	10	2
41	Tetra	500	4	3
42	Three_by_three	500	9	2
43	Three_circles	500	3	2
44	Torus_and_rod	500	2	3
45	Two_diamonds	500	2	2
46	Two_u	260	2	2
47	Wingnut	500	2	2
48	Worms	500	4	2
49	Xor	500	4	2
50	Xor_big_and_small	500	4	2
51	Xor_different_cardinalities	500	4	2
52	Yin_yang	515	4	2

- COP-kmeans - improved (`COPI`) – this method differs to the original by comparing the new and old labels between iterations, rather than comparing the new and old means

- Constrained Spectral Clustering (`CSP`) — This method extends traditional spectral clustering by incorporating prior knowledge through constraints. By embedding these constraints into the spectral embedding process, the algorithm modifies the similarity graph or Laplacian matrix to honour the specified restrictions, thereby steering the clustering towards more informative and accurate partitions. Combining spectral clustering's capability to identify complex cluster structures with the advantages of semi-supervised learning, it enhances clustering performance when partial label information is present. As a non-centroid-based approach, it is particularly well-suited to the task addressed in our study.

- Constrained Average Linkage (`CAL`) – Calculating the distance between two clusters as the average distances between all pairs of data points, one from each cluster.

Table 6.5: Details of the Real datasets.

Dataset	N	C	d
Abalone	4177	3	8
Acute-inflammation	120	2	6
Acute-nephritis	120	2	6
Adult	48842	2	14
Annealing	850	4	31
Arrhythmia	295	10	262
Balance-scale	576	3	4
Bank	4521	2	16
Blood	748	2	4
Breast-cancer-wisc-diag	569	2	30
Breast-cancer-wisc	699	2	9
Breast-cancer	286	2	9
Car	1728	4	6
Cardiotocography-10clases	2126	10	21
Cardiotocography-3clases	2126	3	21
Chess-krvk	28029	18	6
Chess-krvkp	3196	2	36
Congressional-voting	435	2	16
Conn-bench-sonar-mines-rocks	208	2	60
Conn-bench-vowel-deterding	990	11	11
Connect-4	67557	2	42
Contrac	1473	3	9
Credit-approval	690	2	15
Cylinder-bands	512	2	35
Dermatology	297	5	34
Ecoli	272	3	7
Energy-y1	768	3	8
Energy-y2	768	3	8
Glass	146	2	9
Haberman-survival	306	2	3
Hayes-roth	129	2	3
Heart-cleveland	219	2	13
Heart-hungarian	294	2	12
Heart-va	107	2	12
Hill-valley	1212	2	100
Horse-colic	368	2	25
Ilpd-indian-liver	583	2	9
Image-segmentation	2310	7	18
Ionosphere	351	2	33
Iris	150	3	4
Led-display	1000	10	7
Letter	20000	26	16
Low-res-spect	469	4	100
Lymphography	142	3	18
Magic	19020	2	10
Mammographic	961	2	5
Molec-biol-promoter	106	2	57
Molec-biol-splice	3190	3	60
Monks-1	556	2	6
Monks-2	601	2	6
Monks-3	554	2	6
Mushroom	8124	2	21
Musk-1	476	2	166
Musk-2	6598	2	166
Nursery	12958	5	8
Oocytes_merluccius_nucleus_4d	1022	2	41
Oocytes_merluccius_states_2f	1022	3	25
Oocytes_trisopterus_nucleus_2f	912	2	25
Oocytes_trisopterus_states_5b	898	3	32
Optical	5620	10	62
Ozone	2536	2	72
Page-blocks	5445	5	10
Pendigits	10992	10	16
Pima	768	2	8
Planning	182	2	12
Ringnorm	7400	2	20
Seeds	210	3	210
Semeion	1593	10	256
Soybean	362	15	35
Spambase	4601	2	57
Spect	265	2	22
Spectf	267	2	44
Statlog-australian-credit	690	2	14
Statlog-german-credit	1000	2	24
Statlog-heart	270	2	13
Statlog-image	2310	7	18
Statlog-landsat	6435	6	36
Statlog-shuttle	57977	5	9
Statlog-vehicle	846	4	18
Steel-plates	1941	7	27
Synthetic-control	600	6	60
Teaching	102	3	5
Thyroid	7200	3	21
Tic-tac-toe	958	2	9
Titanic	2201	2	3
Twonorm	7400	2	20
Vertebral-column-2clases	310	2	6
Vertebral-column-3clases	310	3	6
Wall-following	5456	4	24
Waveform-noise	5000	3	40
Waveform	5000	3	21
Wine-quality-red	1571	5	11
Wine-quality-white	4873	6	11
Wine	130	2	13
Yeast	1350	5	8

- Constrained Complete Linkage (CCL) – Calculates the distance between two clusters as the maximum distance between a pair of data points, one from each cluster.

- Constrained Single Linkage (CSL) – Calculates the distance between two clusters as the minimum distance between a pair of data points, one from each cluster.

- Constrained Clustering Ensemble (CCEN) – The proposed method

It should be noted that we deliberately excluded certain recent and successful constrained clustering methods from our comparisons. Although 3SHACC [76] falls within the hierarchical clustering category, it proved to be too complex and computationally intensive to be feasible within our experimental framework. Our primary objective is to identify a fast and straightforward method suitable as a potential candidate for future online constrained clustering applications. Additionally, we excluded centroid-based methods such as PCCC [20], recognising that while they may perform effectively on spherical datasets, such as those in our Real data collection, they fall outside the scope of this study.

Experimental Protocol

The experiment was conducted separately on both the synthetic and real datasets. For each dataset, 45 sets of constraints were constructed. Specifically, the number of constraints was varied as a proportion of the total number of objects, N. The proportions considered were $[0, 1, 2, 3, 4, 5, 10, 15, 20]\%$. For a given proportion P^C, the number of constraints, N^C, was calculated using the equation 6.1:

$$N^C = \frac{z(z-1)}{2}, \quad \text{where} \quad z = \text{round}\left(\frac{N \times P^C}{100}\right). \tag{6.1}$$

After determining the number of constraints, we generated N^C unique pairs of points $(P_i, P_j)_1, \ldots, (P_i, P_j)_{N^C}$, ensuring that $i \neq j$. The nature of each constraint was established by comparing the ground-truth labels of P_i and P_j. If both points shared the same label, the pair was designated as a ML constraint. Otherwise, it was classified as a CL constraint.

To account for the randomness inherent in constraint selection, five distinct sets of N^C constraints were generated for each value of P^C. The results were averaged across these five sets to yield a single value for both NMI and ARI for each constraint proportion. Subsequently, these values were averaged over all datasets within the corresponding synthetic or real data groups, thereby providing an overall performance metric for each method at each constraint proportion.

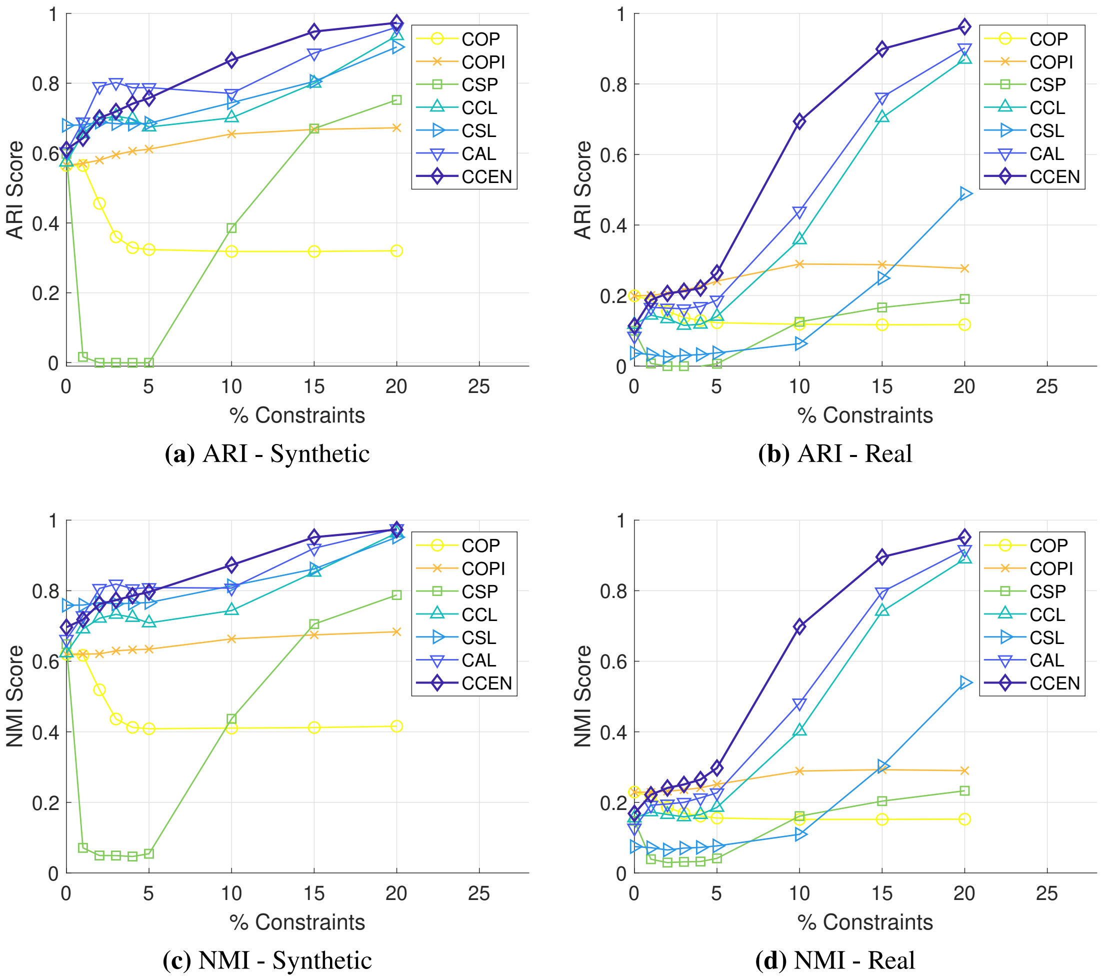
6.2.3 Results

Figure 6.3 presents the ARI and NMI scores for each method across varying proportions of constraints, averaged over both the synthetic and real datasets. From these plots, it is evident that the simpler methods—`CAL`, `CCL`, and `CSL`—consistently demonstrate superior performance according to both metrics, irrespective of the number of constraints provided, particularly in the case of synthetic data. This indicates that these methods are not only effective in accurately assigning data points to the correct clusters, but also proficient in preserving the complex structures inherent in the data. Although the quantity of constraints does not determine whether simpler or more complex methods perform better on synthetic datasets, the performance of the simpler methods improves notably as more constraints are made available.

In contrast, for the real datasets, there is no clear distinction between simpler and more complex methods at lower levels of constraint availability, with some complex methods outperforming certain simpler ones and vice versa. However, as the number of constraints increases, the simpler methods begin to substantially outperform the more complex ones. This is reflected in their improved ability to correctly associate pairs of data points with the ground truth and to maintain the underlying data structures.

Overall, these results underscore both the efficacy of the simpler methods and the influence of constraint quantity on clustering performance. Moreover, the proposed method, `CCEN`, further enhances the performance of the simpler methods across most constraint levels, particularly when applied to real-world datasets.

Table 6.6 presents the execution times for each method on the synthetic and real datasets, respectively. It is evident that the simpler clustering methods are capable of producing partitions significantly faster than their more complex counterparts, with execution



(a) ARI - Synthetic

(b) ARI - Real

(c) NMI - Synthetic

(d) NMI - Real

Figure 6.3: Illustrates the metric scores of the experimental methods for various values of P^C, averaged across all datasets. Each method is represented by a unique colour and marker, as indicated in the plot legend. [189]

times remaining largely unaffected by the number of constraints applied. This highlights not only their superiority in terms of computational efficiency but also their stability to increases in constraint volume. As previously observed, the number of constraints has a direct impact on clustering performance—more constraints generally yield better results. Therefore, the ability of these simpler methods to efficiently incorporate additional constraints without incurring additional computational cost is particularly advantageous.

Furthermore, it is noteworthy that the proposed method, CCEN, does not require substantially more time to execute compared to its base method. This makes CCEN a favourable choice for real-time or time-sensitive applications, as it balances improved clustering performance with computational efficiency.

Table 6.6: Shows the execution time of each method in relation to the proportion of constraints (in milliseconds), averaged across repetitions and datasets. Each cell is colour-coded to indicate speed, with purple representing the fastest methods and orange representing the slowest.[189]

(a) Synthetic Data

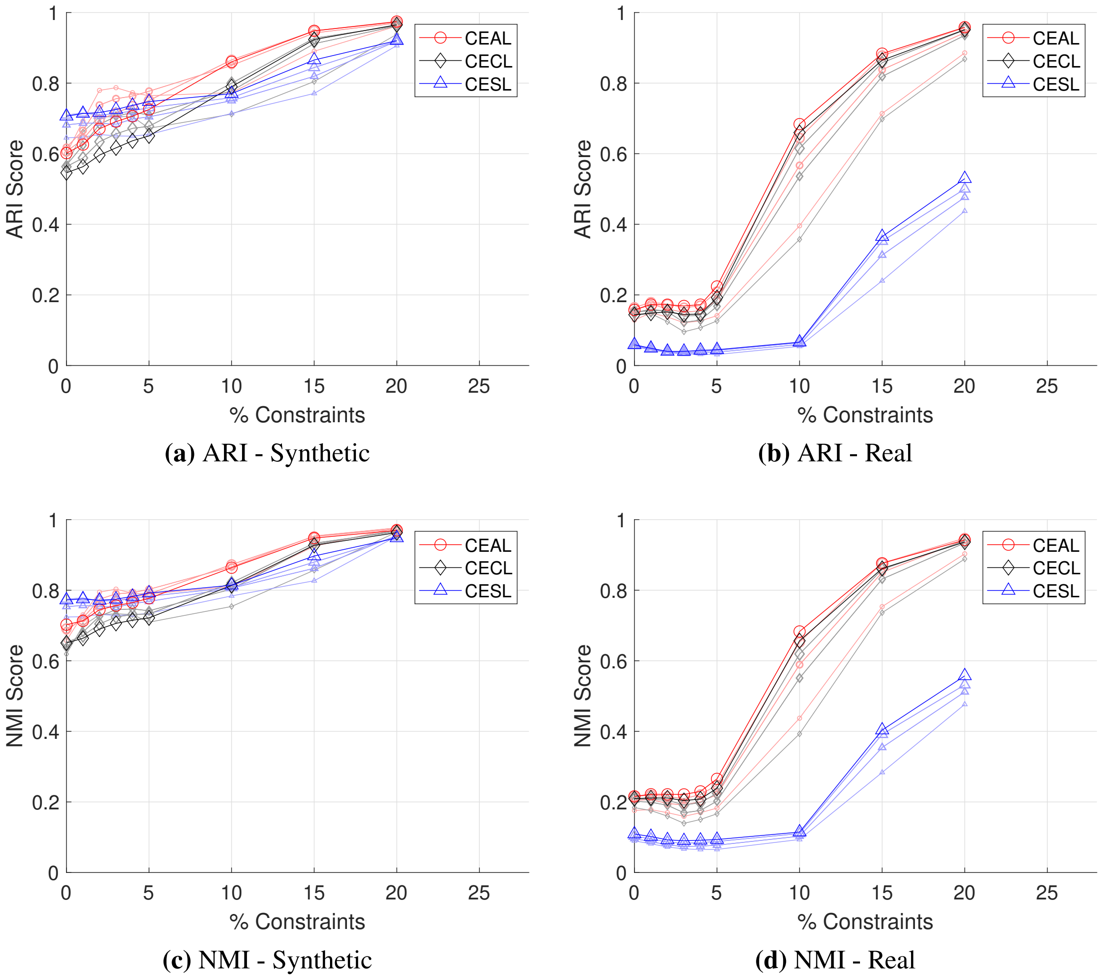
P^C	COP	COPI	CSP	CCL	CSL	CAL	CCEN
0	32	23	1164	8	6	5	12
1	33	29	1051	6	6	5	10
2	29	52	891	6	6	5	10
3	17	88	803	5	6	5	10
4	15	137	834	5	6	5	10
5	13	182	946	5	6	5	10
10	11	474	1254	6	6	6	14
15	11	832	1256	6	6	6	17
20	12	1317	1264	6	6	6	18

(b) Real Data

P^C	COP	COPI	CSP	CCL	CSL	CAL	CCEN
0	40	39	6624	17	17	16	33
1	56	64	6285	17	18	17	28
2	28	166	6165	17	18	16	28
3	17	342	6403	17	18	16	28
4	15	578	6982	17	18	16	30
5	15	803	7300	17	18	17	32
10	12	2566	7471	18	17	18	39
15	13	5988	7473	18	17	18	41
20	15	10729	7499	18	18	18	41

Additionally, we conducted an experimental study to examine whether increasing the number of clusterers in the ensemble has a significant impact on performance. Specifically, we compared the three hierarchical variants as base clusterers (CCBM) for the ensemble. The experimental setup mirrored that described earlier, using both the synthetic and real datasets. The results are presented in Figure 6.4. Each plot depicts a clustering quality metric as a function of the proportion of constraints, for ensembles of four different sizes: 1, 2, 4, and 6 clusterers. Smaller ensembles are represented with smaller markers, whereas the largest ensemble is indicated with the largest markers. The best-performing ensemble is highlighted using more prominent colours. All three methods are displayed on the same plots to enable a direct visual comparison of their performance as base clustering approaches.

The results demonstrate that larger ensembles generally outperform the single-clusterer case (i.e., an ensemble of size 1), indicating that CCEN tends to be more effective than any individual constrained clustering method. The only exception occurs at low constraint levels in the synthetic datasets (Figures 6.4a and 6.4c), where smaller ensembles achieved slightly better performance. Furthermore, the performance difference between ensembles of sizes 4 and 6 is marginal, suggesting that relatively small ensembles can offer a favourable balance between accuracy and computational efficiency, particularly in real-time clustering scenarios.



(a) ARI - Synthetic **(b)** ARI - Real

(c) NMI - Synthetic **(d)** NMI - Real

Figure 6.4: Illustrates the metric scores for the ensemble method using different base clusterers, each represented by a distinct colour as indicated in the plot legend, and varying numbers of ensemble members with $L = [1, 2, 4, 6]$ clusterers. Ensembles with fewer members are indicated by smaller markers, while the best-performing ensemble for each base clusterer is highlighted using a more prominent colour.[189]

Another noteworthy observation is that CEAL consistently outperforms the other ensemble variants. This is consistent with earlier findings, where CAL demonstrated superior performance relative to CCL and CSL, as shown in Figure 6.3.

6.3 Summary

In this chapter, two offline constrained clustering methods—CBC and CCEN—were introduced, with the overarching objective of achieving dependable performance in animal re-identification within an offline context. By utilising information provided by multi-object tracking (MOT) algorithms, such as track labels and frame indices, in conjunction with feature representations extracted via a feature extractor, we developed a method that demonstrated superior performance relative to existing approaches, through a process referred to as classification-based clustering.

However, the experimental results indicated that applying clustering methods to an entire animal video dataset did not yield high re-identification accuracy. This limitation is attributed to the structural complexity inherent in the datasets when considered in their entirety.

Following these findings, together with the results outlined in Section 5.1, it became evident that the approach with the greatest potential—even in an offline setting—would be to reduce the structural complexity of the data through a windowed processing strategy. This approach results in the formation of inherently string-shaped clusters, which are more easily recognised by clustering algorithms. Consequently, a constrained clustering ensemble approach was developed and evaluated on datasets of varying structural complexity. This method demonstrated strong clustering accuracy, which improved further with an increasing number of constraints.

Chapter 7

Conclusion

This chapter summarises and concludes the research done throughout this thesis.

7.1 Summary

Objective: Create a comprehensive dataset encompassing a diverse range of animal species.

Contribution 1 (Presented in Sections 3.1- 3.4): A multi-species animal benchmark dataset was created through a collaborative effort from Bangor University and the University of Burgos, Spain. The set consists of five annotated, unique, and unconstrained video recordings. Each video has five distinct feature representations: Autoencoder (AE), Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), MobileNetV2 (MN2), and RGB Moments (RGB). A detailed analysis was conducted on each video to illustrate the specific challenges and characteristics associated with developing animal re-identification systems. An experimental evaluation of the feature representations was then undertaken within the context of the re-identification task. Results indicated that RGB-based features consistently outperformed the alternatives, emerging as the most effective representation for identifying animals across multiple species. These datasets were subsequently employed to evaluate the performance of all proposed animal re-identification methodologies.

Related Publications:

- L. I. Kuncheva, F. Williams, S. L. Hennessey, and J. J. Rodríguez, "A benchmark database for animal re-identification and tracking," in 2022 IEEE 5th International

Conference on Image Processing Applications and Systems (IPAS), 2022, pp. 1–6.

- L. I. Kuncheva, J. L. Garrido-Labrador, I. Ramos-Pérez, S. L. Hennessey, and J. J. Rodríguez, ”An experiment on animal re-identification from video.” Ecological Informatics, 2023, 74, p.101994.

Objective: Maximise the performance of object detection to reduce the complexity of subsequent animal re-identification.

Contribution 2 (Presented in Sections 4.1.3 & 4.2): A combinatorial approach was proposed for integrating the outputs of both object detection and MOT algorithms. This integration was accomplished through the construction of an adjacency matrix based on IoU thresholding between all pairs of BBs generated by both methods. The resulting adjacency matrix was subsequently utilised to identify the connected components of the corresponding graph, from which the final set of bounding boxes was derived. Experimental results demonstrated that the combined approach outperformed each individual method across all videos in the benchmark dataset.

Related Publications:

- F. J. Williams, L. I. Kuncheva, J. J. Rodríguez, and S. L. Hennessey, “Combination of object tracking and object detection for animal recognition,” in 2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS), 2022, pp. 1–6.

Objective: Investigate the effectiveness of various clustering techniques applied to datasets with complex spatial structure.

Contribution 3 (Presented in Section 5.1.3): An experimental study was undertaken to evaluate suitable clustering techniques for animal video data. Both hierarchical and centroid-based approaches were examined using consecutive temporal windows of varying sizes, enabling the detection and subsequent clustering of prominent spatial patterns within temporally segmented data. This windowed processing framework proved particularly advantageous in revealing local structures that might otherwise be obscured in a global analysis. The findings indicated that the dominant spatial relationships exhibited strong linkage characteristics, thereby supporting the use of hierarchical clustering methods as the most appropriate for this type of data. Furthermore, the results demonstrated that smaller window sizes facilitated more accurate detection and clustering of latent structural patterns, thereby highlighting the potential suitability of processing the data in an online manner.

Related Publications:

- S. L. Hennessey, F. J. Williams, and L. I. Kuncheva, "Hierarchical Vs Centroid-Based Constraint Clustering for Animal Video Data," in 2024 IEEE 12th International Conference on Intelligent Systems (IS), 2024, pp. 1–6. *(Winner of the Best Paper Award)*

Objective: Develop online constraint-based clustering methods to support real-time animal re-identification.

Contribution 4 (Presented in Sections 5.2.1-5.2.5): An online constrained clustering approach (RTSI-ReID) was proposed to process videos of animals on a frame-by-frame basis. Each frame served to re-identify previously observed individuals while also detecting those not yet encountered. Clusters were summarised using key functional statistics. The distinction between re-identification and the recognition of novel individuals was achieved via a likelihood thresholding mechanism, complemented by the integration and ongoing application of instance-level constraints maintained across frames.

To address concept drift, newly added data points influenced the position of the cluster centroid, drawing it towards the new observation. The extent of this influence was governed by a tunable function parameter, enabling adaptability to different application contexts. A forgetting mechanism was intentionally omitted to maintain continuity in identity representation over time.

Experimental results demonstrated that the proposed method outperformed baseline techniques as well as a state-of-the-art online person re-identification algorithm across all five videos in the benchmark dataset. A subsequent ablation study evaluated the contribution of individual components within the algorithm. While two variants exhibited comparable performance, the less complex of the two was selected as the final proposed solution.

Related Publications:

- S. L. Hennessey, F. J. Williams, and L. I. Kuncheva, "Real-Time Online Animal Re-Identification from Video using Spatio-temporal Constraints" (*Under Review in Ecological Informatics)*)

Objective: Develop offline constraint-based clustering methods tailored to the task of animal re-identification.

Contribution 5 (Presented in Sections 6.2.1-6.2.3): An offline constrained clustering ensemble (CCEN) was proposed and evaluated using both real-world and synthetic datasets, encompassing a wide range of cluster shapes and varying levels of clustering difficulty. The method constructed a library of base partitions using a single constrained hierarchical clustering algorithm. The resulting dendrogram was subsequently cut at multiple levels to produce base partitions, each corresponding to a different number of clusters. A cumulative adjacency matrix was then constructed, from which the final partition was derived by extracting the connected components of the associated graph.

Experimental results demonstrated that the proposed method consistently outperformed existing constrained clustering algorithms across all datasets. Furthermore, performance was shown to improve with an increased number of constraints. A follow-up experimental study was conducted to examine the impact of ensemble size and the choice of base clusterers. The findings revealed that only a small ensemble was required to achieve efficient and robust performance. Among the evaluated base clusterers, constrained hierarchical clustering with average linkage yielded the best results and was therefore adopted as the default base method.

Related Publications:

- F. J. Williams, S. L. Hennessey, L. I. Kuncheva, J. F. Diez-Pastor, and J. J. Rodríguez, "A Constrained Cluster Ensemble Using Hierarchical Clustering

Methods," in 2024 IEEE 12th International Conference on Intelligent Systems (IS), 2024, pp. 1–6.

Contribution 6 (Presented in Sections 6.1.1-6.1.3): A classification-based clustering (CBC) approach was proposed and evaluated using both unconstrained and constrained animal video datasets. The method leverages the outputs of tracking algorithms—specifically, the labelled tracks—to train a classifier, which is subsequently used to reclassify the raw data via resubstitution. A confusion matrix was then constructed, implicitly encoding instance-level CL constraints. This matrix was row-normalised, and the entry with the highest value was identified, corresponding to the pair of tracks exhibiting the greatest similarity. These tracks were then merged. This procedure was repeated iteratively, enabling hierarchical merging of tracks until a predefined number of clusters was obtained.

An experimental study was conducted to evaluate various post-clustering strategies and different tracking methods. The results demonstrated that the classification-based clustering approach, which fully incorporated information derived from the tracking process, achieved the best performance across all datasets. Additionally, the study highlighted the effectiveness of even relatively simple track types in supporting the clustering process.

Related Publications:

- F. J. Williams, S. L. Hennessey, L. I. Kuncheva, "Animal Re-Identification in Video through Track Clustering" Pattern Analysis and Applications 28, no. 3 (2025): 125.

7.2 Future Work

Future work in the field of species-invariant animal re-identification remains extensive, as the area is still in its early stages of development. Nonetheless, one of the most compelling and challenging directions—particularly from my perspective—is the construction of a universal feature representation capable of distinguishing individual animals across all species. The following are several proposed avenues for future research aimed at advancing towards this overarching objective:

- Traditional feature descriptors are often designed for rectangular regions, primarily due to the widespread adoption of BB extraction algorithms. Consequently, these descriptors typically produce a fixed number of features, regardless of the size of the BB. However, such BBs may compromise the quality of the resulting feature representations, as they frequently include background elements and, in some instances, additional animals. To address this limitation, image segmentation techniques can be employed to isolate the target animal, thereby removing irrelevant information and enhancing the clarity of the representation. Building on this approach, there is a clear need to develop traditional feature extraction algorithms specifically designed for arbitrarily shaped image segments. These algorithms must be capable of generating a consistent number of features—independent of the segment's shape or size—while retaining their capacity to distinguish between individual animals.

- Exploring the integration of simple feature descriptors that capture keypoints, shape, texture, and colour—such as SIFT, LBP, HOG, and RGB—with image segmentation methods to construct a more universal and generalisable species-invariant feature representation. When combined with autonomously generated instance-level constraints, this approach has the potential to enhance the discrimination of individual animals across multiple species. Furthermore, it may facilitate unsupervised learning by reducing the structural complexity of the data, thereby improving the overall accuracy and scalability of species-invariant animal re-identification systems.

References

[1] W. Abbas, D. Masip and A. Giovannucci, 'Limbs detection and tracking of head-fixed mice for behavioral phenotyping using motion tubes and deep learning,' *IEEE Access*, vol. 8, pp. 37 891–37 901, 2020 (p. 12).

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, 'Slic superpixels compared to state-of-the-art superpixel methods,' *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012 (p. 21).

[3] L. Adam, V. Čermák, K. Papafitsoros and L. Picek, 'Seaturtleid2022: A long-span dataset for reliable sea turtle re-identification,' in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 7146–7156 (p. 16).

[4] C. C. Aggarwal, S. Y. Philip, J. Han and J. Wang, 'A framework for clustering evolving data streams,' in *Proceedings 2003 VLDB conference*, Elsevier, 2003, pp. 81–92 (pp. 23, 24).

[5] S. Ahmed, T. Gaber, A. Tharwat, A. E. Hassanien and V. Snáel, 'Muzzle-based cattle identification using speed up robust feature approach,' in *2015 International Conference on Intelligent Networking and Collaborative Systems*, IEEE, 2015, pp. 99–104 (pp. 12, 14).

[6] A. Allen, B. Golden, M. Taylor, D. Patterson, D. Henriksen and R. Skuce, 'Evaluation of retinal imaging technology for the biometric identification of bovine animals in northern ireland,' *Livestock science*, vol. 116, no. 1-3, pp. 42–52, 2008 (p. 12).

[7] M. Alziati, F. Amarù, L. Magri and F. Arrigoni, 'Ensemble clustering via synchronized relabelling,' *Pattern Recognition Letters*, vol. 184, pp. 176–182, 2024 (p. 26).

[8] W. Andrew, 'Visual biometric processes for collective identification of individual friesian cattle,' Ph.D. dissertation, University of Bristol, 2019 (p. 16).

[9] W. Andrew, J. Gao, S. Mullan, N. Campbell, A. W. Dowsey and T. Burghardt, ‘Visual identification of individual holstein-friesian cattle via deep metric learning,’ *Computers and Electronics in Agriculture*, vol. 185, p. 106 133, 2021 (p. 15).

[10] W. Andrew, C. Greatwood and T. Burghardt, ‘Visual localisation and individual identification of holstein friesian cattle via deep learning,’ in *Proceedings of the IEEE international conference on computer vision workshops*, 2017, pp. 2850–2859 (pp. 11, 13, 14).

[11] W. Andrew, C. Greatwood and T. Burghardt, ‘Aerial animal biometrics: Individual friesian cattle recovery and visual identification via an autonomous uav with onboard deep inference,’ in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 237–243 (p. 13).

[12] S. W. Arachchilage and E. Izquierdo, ‘Adaptive aggregated tracklet linking for multi-face tracking,’ in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, pp. 1366–1370 (p. 22).

[13] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez and I. Perona, ‘An extensive comparative study of cluster validity indices,’ *Pattern recognition*, vol. 46, no. 1, pp. 243–256, 2013 (p. 46).

[14] R. Ardekani et al., ‘Three-dimensional tracking and behaviour monitoring of multiple fruit flies,’ *Journal of The Royal Society Interface*, vol. 10, no. 78, p. 20 120 547, 2013 (p. 28).

[15] A. Ardovini, L. Cinque and E. Sangineto, ‘Identifying elephant photos by multi-curve matching,’ *Pattern Recognition*, vol. 41, no. 6, pp. 1867–1877, 2008 (pp. 11, 12, 14).

[16] D. Arthur and S. Vassilvitskii, ‘K-means++: The advantages of careful seeding,’ Stanford, Tech. Rep., 2006 (p. 77).

[17] A. I. Awad, ‘From classical methods to animal biometrics: A review on cattle identification and tracking,’ *Computers and Electronics in Agriculture*, vol. 123, pp. 423–435, 2016 (p. 14).

[18] U. G. Barron, G. Corkery, B. Barry, F. Butler, K. McDonnell and S. Ward, ‘Assessment of retinal recognition technology as a biometric method for sheep identification,’ *Computers and electronics in agriculture*, vol. 60, no. 2, pp. 156–166, 2008 (p. 12).

[19] S. Basu, A. Banerjee and R. J. Mooney, 'Active semi-supervision for pairwise constrained clustering,' in *Proceedings of the 2004 SIAM international conference on data mining*, SIAM, 2004, pp. 333–344 (pp. 22, 23).

[20] P. Baumann and D. S. Hochbaum, 'Pccc: The pairwise-confidence-constraints-clustering algorithm,' *arXiv preprint arXiv:2212.14437*, 2022 (pp. 76, 125).

[21] T. L. Berg and D. A. Forsyth, 'Animals on the web,' in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, IEEE, vol. 2, 2006, pp. 1463–1470 (p. 15).

[22] L. Bergamini et al., 'Multi-views embedding for cattle re-identification,' in *2018 14th international conference on signal-image technology & internet-based systems (SITIS)*, IEEE, 2018, pp. 184–191 (pp. 13, 14).

[23] L. Bergamini et al., 'Extracting accurate long-term behavior changes from a large pig dataset,' in *16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2021*, SciTePress, 2021, pp. 524–533 (p. 108).

[24] C. Bergler et al., 'Fin-print a fully-automated multi-stage deep-learning-based framework for the individual recognition of killer whales,' *Scientific reports*, vol. 11, no. 1, p. 23 480, 2021 (p. 16).

[25] J. C. Bezdek and J. M. Keller, 'Streaming data analysis: Clustering or classification?' *IEEE transactions on systems, man, and cybernetics: systems*, vol. 51, no. 1, pp. 91–102, 2020 (p. 83).

[26] M. Bigg, 'An assessment of killer whale (orcinus orca) stocks off vancouver island, british columbia,' *Report of the International Whaling Commission*, vol. 32, no. 65, pp. 655–666, 1982 (p. 10).

[27] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4 (pp. 18, 19).

[28] J. Blancou, 'A history of the traceability of animals and animal products.,' *Revue scientifique et technique (International Office of Epizootics)*, vol. 20, no. 2, pp. 413–425, 2001 (p. 9).

[29] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, 'Yolov4: Optimal speed and accuracy of object detection,' *arXiv preprint arXiv:2004.10934*, 2020 (p. 28).

[30] P. Bodesheim, J. Blunk, M. Koerschens, C.-A. Brust, C. Kaeding and J. Denzler, 'Pre-trained models are not enough: Active and lifelong learning is important for long-term visual monitoring of mammals in biodiversity research—individual

identification and attribute prediction with image features from deep neural networks and decoupled decision models applied to elephants and great apes,' *Mammalian Biology*, vol. 102, no. 3, pp. 875–897, 2022 (p. 13).

[31] T. Boongoen and N. Iam-On, 'Cluster ensembles: A survey of approaches with recent extensions and applications,' *Computer Science Review*, vol. 28, pp. 1–25, 2018 (p. 119).

[32] S. Bouma, M. D. Pawley, K. Hupman and A. Gilman, 'Individual common dolphin identification via metric embedding learning,' in *2018 international conference on image and vision computing New Zealand (IVCNZ)*, IEEE, 2018, pp. 1–6 (p. 12).

[33] O. Brookes and T. Burghardt, 'A dataset and application for facial recognition of individual gorillas in zoo environments,' *arXiv preprint arXiv:2012.04689*, 2020 (pp. 13, 17).

[34] A. Brown, V. Kalogeiton and A. Zisserman, *Face, body, voice: Video person-clustering with multiple modalities*, 2021. arXiv: `2105.09939 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2105.09939` (p. 106).

[35] J. Bruslund Haurum, A. Karpova, M. Pedersen, S. Hein Bengtson and T. B. Moeslund, 'Re-identification of zebrafish using metric learning,' in *Proceedings of the IEEE/CVF winter conference on applications of computer vision workshops*, 2020, pp. 1–11 (p. 16).

[36] P. Buehler, B. Carroll, A. Bhatia, V. Gupta and D. E. Lee, 'An automated program to find animals and crop photographs for individual recognition,' *Ecological informatics*, vol. 50, pp. 191–196, 2019 (p. 12).

[37] T. Burghardt and J. Calic, 'Analysing animal behaviour in wildlife videos using face detection and tracking,' *IEE Proceedings-Vision, Image and Signal Processing*, vol. 153, no. 3, pp. 305–312, 2006 (p. 11).

[38] T. Burghardt and N. Campbell, 'Individual animal identification using visual biometrics on deformable coat patterns,' in *International Conference on Computer Vision Systems: Proceedings (2007)*, 2007 (pp. 11, 13).

[39] T. Burghardt, B. Thomas, P. J. Barham and J. Calic, 'Automated visual recognition of individual african penguins,' in *Fifth International Penguin Conference*, 2004 (pp. 11, 12).

[40] F. Cao, M. Estert, W. Qian and A. Zhou, 'Density-based clustering over an evolving data stream with noise,' in *Proceedings of the 2006 SIAM international conference on data mining*, SIAM, 2006, pp. 328–339 (p. 24).

[41] X. Cao, C. Zhang, C. Zhou, H. Fu and H. Foroosh, 'Constrained multi-view video face clustering,' *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4381–4393, 2015 (p. 22).

[42] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov and S. Zagoruyko, 'End-to-end object detection with transformers,' in *European conference on computer vision*, Springer, 2020, pp. 213–229 (p. 28).

[43] J. Chan, H. Carrión, R. Mégret, J. L. Agosto-Rivera and T. Giray, 'Honeybee re-identification in video: New datasets and impact of self-supervision.,' in *VISIGRAPP (5: VISAPP)*, 2022, pp. 517–525 (p. 13).

[44] V. Chandola, A. Banerjee and V. Kumar, 'Anomaly detection: A survey,' *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009 (p. 21).

[45] T. Cheeseman, T. Johnson, K. Southerland and N. Muldavin, 'Happywhale: Globalizing marine mammal photo identification via a citizen science web platform,' *Happywhale, Santa Cruz, CA, USA, Rep. SC/67b/PH/02*, 2017 (p. 16).

[46] T. Cheeseman et al., 'Advanced image recognition: A fully automated, high-accuracy photo-identification matching system for humpback whales,' *Mammalian Biology*, vol. 102, no. 3, pp. 915–929, 2022 (p. 16).

[47] C. Chen, A. Seff, A. Kornhauser and J. Xiao, 'Deepdriving: Learning affordance for direct perception in autonomous driving,' in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2722–2730 (p. 19).

[48] D. Chen, Y. Yang, H. Wang and A. Mahmood, 'Convergence analysis of semi-supervised clustering ensemble,' in *2013 IEEE Third International Conference on Information Science and Technology (ICIST)*, IEEE, 2013, pp. 783–788 (p. 27).

[49] G. Chen, T. X. Han, Z. He, R. Kays and T. Forrester, 'Deep convolutional neural network based species recognition for wild animal monitoring,' in *2014 IEEE international conference on image processing (ICIP)*, IEEE, 2014, pp. 858–862 (p. 14).

[50] K. Chen et al., 'Mmdetection: Open mmlab detection toolbox and benchmark,' *arXiv preprint arXiv:1906.07155*, 2019 (p. 59).

[51] Z. Chen et al., ‘Alphatracker: A multi-animal tracking and behavioral analysis tool,’ *Frontiers in Behavioral Neuroscience*, vol. 17, p. 1 111 908, 2023 (p. 29).

[52] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri and F. Herrera, ‘Deep learning in video multi-object tracking: A survey,’ *Neurocomputing*, vol. 381, pp. 61–88, 2020 (p. 28).

[53] J. P. Crall, C. V. Stewart, T. Y. Berger-Wolf, D. I. Rubenstein and S. R. Sundaresan, ‘Hotspotter—patterned species instance recognition,’ in *2013 IEEE workshop on applications of computer vision (WACV)*, IEEE, 2013, pp. 230–237 (pp. 12–14).

[54] D. Crouse et al., ‘Lemurfaceid: A face recognition system to facilitate individual identification of lemurs,’ *Bmc Zoology*, vol. 2, no. 1, p. 2, 2017 (pp. 11, 13).

[55] I. Davidson and S. Basu, ‘A survey of clustering with instance level constraints,’ *ACM Transactions on Knowledge Discovery from data*, vol. 1, no. 1-41, pp. 2–42, 2007 (p. 21).

[56] D. L. Davies and D. W. Bouldin, ‘A cluster separation measure,’ *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 2009 (pp. 25, 47).

[57] D. Deb et al., ‘Face recognition: Primates in the wild,’ in *2018 IEEE 9th international conference on biometrics theory, applications and systems (BTAS)*, IEEE, 2018, pp. 1–10 (pp. 12, 13).

[58] D. Dinler and M. K. Tural, ‘A survey of constrained clustering,’ in *Unsupervised learning algorithms*, Springer, 2016, pp. 207–235 (p. 21).

[59] G. Ditzler, M. Roveri, C. Alippi and R. Polikar, ‘Learning in nonstationary environments: A survey,’ *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, 2015 (p. 44).

[60] N. Dlamini and T. L. Van Zyl, ‘Comparing class-aware and pairwise loss functions for deep metric learning in wildlife re-identification,’ *Sensors*, vol. 21, no. 18, p. 6109, 2021 (pp. 12, 13).

[61] E. Dobriban and S. Wager, ‘High-dimensional asymptotics of prediction: Ridge regression and classification,’ *The Annals of Statistics*, vol. 46, no. 1, pp. 247–279, 2018 (p. 19).

[62] J. C. Dunn, ‘Well-separated clusters and optimal fuzzy partitions,’ *Journal of cybernetics*, vol. 4, no. 1, pp. 95–104, 1974 (p. 25).

[63] J. Duyck, C. Finn, A. Hutcheon, P. Vera, J. Salas and S. Ravela, 'Sloop: A pattern retrieval engine for individual animal identification,' *Pattern Recognition*, vol. 48, no. 4, pp. 1059–1073, 2015 (p. 13).

[64] W. J. Eradus and M. B. Jansen, 'Animal identification and monitoring,' *Computers and Electronics in Agriculture*, vol. 24, no. 1-2, pp. 91–98, 1999 (p. 14).

[65] M. Ester, H.-P. Kriegel, J. Sander, X. Xu et al., 'A density-based algorithm for discovering clusters in large spatial databases with noise,' in *kdd*, vol. 96, 1996, pp. 226–231 (p. 106).

[66] M. E. Evans, 'Recognizing individual bewick's swans by bill pattern,' *Wildfowl*, vol. 28, no. 28, p. 6, 1977 (pp. 9, 11).

[67] G. Falzon et al., 'Classifyme: A field-scouting software for the identification of wildlife in camera trap images,' *Animals*, vol. 10, no. 1, p. 58, 2019 (p. 14).

[68] A. C. Ferreira et al., 'Deep learning-based methods for individual recognition in small birds,' *Methods in Ecology and Evolution*, vol. 11, no. 9, pp. 1072–1085, 2020 (p. 17).

[69] C. Fraley and A. E. Raftery, 'Model-based clustering, discriminant analysis, and density estimation,' *Journal of the American statistical Association*, vol. 97, no. 458, pp. 611–631, 2002 (p. 20).

[70] A. Freytag, E. Rodner, M. Simon, A. Loos, H. S. Kühl and J. Denzler, 'Chimpanzee faces in the wild: Log-euclidean cnns for predicting identities and attributes of primates,' in *German conference on pattern recognition*, Springer, 2016, pp. 51–63 (pp. 11, 12).

[71] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy and A. Bouchachia, 'A survey on concept drift adaptation,' *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014 (pp. 23, 24).

[72] J. Gao, T. Burghardt, W. Andrew, A. W. Dowsey and N. W. Campbell, 'Towards self-supervision for video identification of individual holstein-friesian cattle: The cows2021 dataset,' *arXiv preprint arXiv:2105.01938*, 2021 (p. 15).

[73] M. Ghesmoune, M. Lebbah and H. Azzag, 'State-of-the-art on clustering data streams,' *Big Data Analytics*, vol. 1, no. 1, p. 13, 2016 (p. 24).

[74] A. Girbau, F. Marqués and S. Satoh, 'Multiple object tracking from appearance by hierarchically clustering tracklets,' *arXiv preprint arXiv:2210.03355*, 2022 (p. 111).

[75] K. Golalipour, E. Akbari, S. S. Hamidi, M. Lee and R. Enayatifar, 'From clustering to clustering ensemble selection: A review,' *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104 388, 2021 (p. 119).

[76] G. Gonzalez-Almagro, J. L. Suárez, J. Luengo, J.-R. Cano and S. García, '3SHACC: Three stages hybrid agglomerative constrained clustering,' *Neurocomputing*, vol. 490, pp. 441–461, 2022. DOI: `https://doi.org/10.1016/j.neucom.2021.12.018` (p. 125).

[77] G. González-Almagro, D. Peralta, E. De Poorter, J.-R. Cano and S. García, 'Semi-supervised constrained clustering: An in-depth overview, ranked taxonomy and future research directions,' *Artificial Intelligence Review*, vol. 58, no. 5, p. 157, 2025 (p. 27).

[78] C. Gope, N. Kehtarnavaz, G. Hillman and B. Würsig, 'An affine invariant curve matching method for photo-identification of marine mammals,' *Pattern Recognition*, vol. 38, no. 1, pp. 125–132, 2005 (pp. 12, 13).

[79] N. Grira, M. Crucianu and N. Boujemaa, 'Unsupervised and semi-supervised clustering: A brief survey,' *A review of machine learning techniques for processing multimedia content*, vol. 1, no. 2004, pp. 9–16, 2004 (p. 22).

[80] H. Habe, Y. Takeuchi, K. Terayama and M.-a. Sakagami, 'Pose estimation of swimming fish using naca airfoil model for collective behavior analysis,' *Journal of Robotics and Mechatronics*, vol. 33, no. 3, pp. 547–555, 2021 (p. 11).

[81] M. Hahsler and M. Bolaños, 'Clustering data streams based on shared density between micro-clusters,' *IEEE transactions on knowledge and data engineering*, vol. 28, no. 6, pp. 1449–1461, 2016 (p. 89).

[82] M. F. Hansen et al., 'Towards on-farm pig face recognition using convolutional neural networks,' *Computers in Industry*, vol. 98, pp. 145–152, 2018 (pp. 13, 14).

[83] P. E. Hart, D. G. Stork and R. Duda, *Pattern classification*. Wiley Hoboken, 2001 (p. 19).

[84] B. A. Hassan, N. B. Tayfor, A. A. Hassan, A. M. Ahmed, T. A. Rashid and N. N. Abdalla, 'From a-to-z review of clustering validation indices,' *Neurocomputing*, vol. 601, p. 128 198, 2024 (p. 46).

[85] K. He, G. Gkioxari, P. Dollár and R. Girshick, 'Mask r-cnn,' in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969 (p. 28).

[86] K. He, X. Zhang, S. Ren and J. Sun, 'Deep residual learning for image recognition,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778 (p. 19).

[87] J. Holmberg, L. Otarashvili and J. Smith, 'Where's whale-do?' *Evaluating competitor approaches to machine learning-based re-ID of belugas. Sterling (VA): US Department of the Interior, Bureau of Ocean Energy Management*, 2024 (p. 16).

[88] H. Hotelling, 'Analysis of a complex of statistical variables into principal components.,' *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933 (p. 41).

[89] J. Hou et al., 'Identification of animal individuals using deep learning: A case study of giant panda,' *Biological Conservation*, vol. 242, p. 108 414, 2020 (pp. 13, 14).

[90] L. Hubert and P. Arabie, 'Comparing partitions,' *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985 (p. 25).

[91] B. Hughes and T. Burghardt, 'Automated visual fin identification of individual great white sharks,' *International Journal of Computer Vision*, vol. 122, pp. 542–557, 2017 (pp. 14, 16).

[92] A. K. Jain, 'Data clustering: 50 years beyond k-means,' *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010 (p. 20).

[93] A. K. Jain, A. Ross and S. Prabhakar, 'An introduction to biometric recognition,' *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004 (p. 18).

[94] Y. Jia, S. Tao, R. Wang and Y. Wang, 'Ensemble clustering via co-association matrix self-enhancement,' *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 11 168–11 179, 2023 (pp. 26, 27).

[95] J. N. Jover et al., 'An automatic colour-based computer vision algorithm for tracking the position of piglets,' *Spanish Journal of Agricultural Research*, vol. 7, no. 3, pp. 535–549, 2009 (pp. 12, 14).

[96] V. Kalogeiton and A. Zisserman, 'Constrained video face clustering using 1nn relations,' 2020 (p. 22).

[97] M. Kang and Y. C. Lim, 'Pedestrian detection using hog-based block selection,' in *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, IEEE, vol. 2, 2014, pp. 783–787 (p. 35).

[98] M. Kashiha et al., ‘Automatic identification of marked pigs in a pen using image pattern recognition,’ *Computers and electronics in agriculture*, vol. 93, pp. 111–120, 2013 (p. 14).

[99] M. Kelly, R. Longjohn and K. Nottingham, *The uci machine learning repository*, `https://archive.ics.uci.edu`, 2024 (p. 121).

[100] Z. Khan, T. Balch and F. Dellaert, ‘Mcmc-based particle filtering for tracking a variable number of interacting targets,’ *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 11, pp. 1805–1819, 2005 (pp. 28, 29).

[101] M. Kim and R. Ramakrishna, ‘New indices for cluster validity assessment,’ *Pattern Recognition Letters*, vol. 26, no. 15, pp. 2353–2363, 2005 (p. 46).

[102] D. Klein, S. D. Kamvar and C. D. Manning, ‘From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering,’ in *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002, pp. 307–314 (pp. 22, 23, 75).

[103] M. Korschens and J. Denzler, ‘Elpephants: A fine-grained dataset for elephant re-identification,’ in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0 (pp. 14, 17).

[104] M. Körschens, B. Barz and J. Denzler, ‘Towards automatic identification of elephants in the wild,’ *arXiv preprint arXiv:1812.04418*, 2018 (p. 17).

[105] H.-P. Kriegel, P. Kröger, J. Sander and A. Zimek, ‘Density-based clustering,’ *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 3, pp. 231–240, 2011 (p. 20).

[106] A. Krizhevsky, I. Sutskever and G. E. Hinton, ‘Imagenet classification with deep convolutional neural networks,’ *Advances in neural information processing systems*, vol. 25, 2012 (p. 19).

[107] H. S. Kühl and T. Burghardt, ‘Animal biometrics: Quantifying and detecting phenotypic appearance,’ *Trends in ecology & evolution*, vol. 28, no. 7, pp. 432–441, 2013 (pp. 11–13).

[108] H. W. Kuhn, ‘The hungarian method for the assignment problem,’ *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955 (p. 86).

[109] S. Kullback and R. A. Leibler, ‘On information and sufficiency,’ *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951 (p. 44).

[110] P. Kulshreshtha and T. Guha, 'An online algorithm for constrained face clustering in videos,' in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 2670–2674 (p. 89).

[111] S. Kumar and S. K. Singh, 'Visual animal biometrics: Survey,' *Iet Biometrics*, vol. 6, no. 3, pp. 139–156, 2017 (p. 11).

[112] S. Kumar and S. K. Singh, 'Monitoring of pet animal in smart cities using animal biometrics,' *Future Generation Computer Systems*, vol. 83, pp. 553–563, 2018 (pp. 11, 13).

[113] S. Kumar, S. Tiwari and S. K. Singh, 'Face recognition for cattle,' in *2015 Third International Conference on Image Information Processing (ICIIP)*, IEEE, 2015, pp. 65–72 (pp. 13, 14).

[114] L. I. Kuncheva, J. L. Garrido-Labrador, I. Ramos-Pérez, S. L. Hennessey and J. J. Rodríguez, 'An experiment on animal re-identification from video,' *Ecological Informatics*, vol. 74, p. 101 994, 2023 (pp. 52, 54–56).

[115] L. I. Kuncheva and D. P. Vetrov, 'Evaluation of stability of k-means cluster ensembles with respect to random initialization,' *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 11, pp. 1798–1808, 2006 (p. 119).

[116] L. I. Kuncheva and C. J. Whitaker, 'Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,' *Machine learning*, vol. 51, no. 2, pp. 181–207, 2003 (p. 26).

[117] M. Lahiri, C. Tantipathananandh, R. Warungu, D. I. Rubenstein and T. Y. Berger-Wolf, 'Biometric animal databases from field photographs: Identification of individual zebra in the wild,' in *Proceedings of the 1st ACM international conference on multimedia retrieval*, 2011, pp. 1–8 (pp. 14, 17).

[118] Y. Lai, S. He, Z. Lin, F. Yang, Q. Zhou and X. Zhou, 'An adaptive robust semi-supervised clustering framework using weighted consensus of random k k-means ensemble,' *IEEE Transactions on Knowledge and data engineering*, vol. 33, no. 5, pp. 1877–1890, 2019 (p. 27).

[119] J. I. Larregui, D. Cazzato and S. M. Castro, 'An image processing pipeline to segment iris for unconstrained cow identification system,' *Open Computer Science*, vol. 9, no. 1, pp. 145–159, 2019 (p. 12).

[120] S. Li, J. Li, H. Tang, R. Qian and W. Lin, 'Atrw: A benchmark for amur tiger re-identification in the wild,' *arXiv preprint arXiv:1906.05586*, 2019 (pp. 14, 17).

[121] S. Li, H. Ren, X. Xie and Y. Cao, 'A review of multi-object tracking in recent times,' *IET Computer Vision*, vol. 19, no. 1, e70010, 2025 (pp. 28, 29).

[122] X. Li et al., 'Video object segmentation with re-identification,' *arXiv preprint arXiv:1708.00197*, 2017 (p. 12).

[123] S. Liao, Y. Hu, X. Zhu and S. Z. Li, 'Person re-identification by local maximal occurrence representation and metric learning,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2197–2206 (p. 19).

[124] G. Litjens et al., 'A survey on deep learning in medical image analysis,' *Medical Image Analysis*, vol. 42, pp. 60–88, 2017 (p. 19).

[125] A. Loos and A. Ernst, 'An automated chimpanzee identification system using face detection and recognition,' *EURASIP Journal on Image and Video Processing*, vol. 2013, pp. 1–17, 2013 (p. 17).

[126] D. G. Lowe, 'Distinctive image features from scale-invariant keypoints,' *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004 (p. 10).

[127] Z. Lu and M. A. Carreira-Perpinan, 'Constrained spectral clustering through affinity propagation,' in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8 (pp. 22, 23).

[128] M. Meilă, 'Comparing clusterings—an information based distance,' *Journal of multivariate analysis*, vol. 98, no. 5, pp. 873–895, 2007 (p. 78).

[129] V. Miele, G. Dussert, B. Spataro, S. Chamaillé-Jammes, D. Allainé and C. Bonenfant, 'Revisiting animal photo-identification using deep metric learning and network analysis,' *Methods in Ecology and Evolution*, vol. 12, no. 5, pp. 863–873, 2021 (p. 17).

[130] N. Murali, J. Schneider, J. Levine and G. Taylor, 'Classification and re-identification of fruit fly individuals across days with convolutional neural networks,' in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, pp. 570–578 (p. 18).

[131] F. Murtagh and P. Contreras, 'Algorithms for hierarchical clustering: An overview,' *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 2, no. 1, pp. 86–97, 2012 (p. 20).

[132] E. Nepovinnykh et al., 'Sealid: Saimaa ringed seal re-identification dataset,' *Sensors*, vol. 22, no. 19, p. 7602, 2022 (p. 16).

[133] A. Ng, M. Jordan and Y. Weiss, 'On spectral clustering: Analysis and an algorithm,' *Advances in neural information processing systems*, vol. 14, 2001 (pp. 20, 106).

[134] H.-L. Nguyen, Y.-K. Woon and W.-K. Ng, 'A survey on data stream clustering and classification,' *Knowledge and information systems*, vol. 45, no. 3, pp. 535–569, 2015 (p. 24).

[135] M. S. Norouzzadeh et al., 'Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning,' *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, E5716–E5725, 2018 (p. 14).

[136] F. Okura, S. Ikuma, Y. Makihara, D. Muramatsu, K. Nakada and Y. Yagi, 'Rgb-d video-based individual identification of dairy cows using gait and texture analyses,' *Computers and Electronics in Agriculture*, vol. 165, p. 104 944, 2019 (p. 13).

[137] V. Panadeiro, A. Rodriguez, J. Henry, D. Wlodkowic and M. Andersson, 'A review of 28 free animal-tracking software applications: Current features and limitations,' *Lab animal*, vol. 50, no. 9, pp. 246–254, 2021 (p. 29).

[138] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf and D. I. Rubenstein, 'Animal population censusing at scale with citizen science and photographic identification,' in *AAAI spring symposium-technical report*, 2017 (p. 17).

[139] J. Parham, C. Stewart, J. Crall, D. Rubenstein, J. Holmberg and T. Berger-Wolf, 'An animal detection pipeline for identification,' in *2018 IEEE winter conference on applications of computer vision (WACV)*, IEEE, 2018, pp. 1075–1083 (pp. 12, 14).

[140] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda and G. G. De Polavieja, 'Idtracker: Tracking individuals in a group by automatic identification of unmarked animals,' *Nature methods*, vol. 11, no. 7, pp. 743–748, 2014 (p. 30).

[141] A. Pesaranghader and H. L. Viktor, 'Fast hoeffding drift detection method for evolving data streams,' in *Joint European conference on machine learning and knowledge discovery in databases*, Springer, 2016, pp. 96–111 (p. 44).

[142] R. Petegrosso, Z. Li and R. Kuang, 'Machine learning and statistical methods for clustering single-cell rna-sequencing data,' *Briefings in bioinformatics*, vol. 21, no. 4, pp. 1209–1223, 2020 (p. 21).

[143] D. Ramanan, D. A. Forsyth and K. Barnard, 'Building models of animals from video,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1319–1334, 2006 (pp. 12, 13).

[144] P. C. Ravoor and T. Sudarshan, 'Deep learning methods for multi-species animal re-identification and tracking–a survey,' *Computer Science Review*, vol. 38, p. 100 289, 2020 (p. 13).

[145] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. Heras and G. G. De Polavieja, 'Idtracker. ai: Tracking all individuals in small or large collectives of unmarked animals,' *Nature methods*, vol. 16, no. 2, pp. 179–182, 2019 (p. 29).

[146] P. J. Rousseeuw, 'Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,' *Journal of Computational and Applied Mathematics*, vol. 20, no. 1, pp. 53–65, 1987 (pp. 25, 47).

[147] F. Sakib and T. Burghardt, 'Visual recognition of great ape behaviours in the wild,' *arXiv preprint arXiv:2011.10759*, 2020 (p. 11).

[148] S. Sarfraz, V. Sharma and R. Stiefelhagen, 'Efficient parameter-free clustering using first neighbor relations,' in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8934–8943 (p. 106).

[149] J. Schneider, N. Murali, G. Taylor and J. Levine, *Dataset for: Can Drosophila melanogaster tell who's who?* Version V1, 2018. DOI: `10.5683/SP2/JP4WDF`. [Online]. Available: `https://doi.org/10.5683/SP2/JP4WDF` (p. 18).

[150] S. Schneider, G. W. Taylor and S. Kremer, 'Deep learning object detection methods for ecological camera trap data,' in *2018 15th Conference on computer and robot vision (CRV)*, IEEE, 2018, pp. 321–328 (pp. 12, 14).

[151] S. Schneider, G. W. Taylor and S. C. Kremer, 'Similarity learning networks for animal individual re-identification-beyond the capabilities of a human observer,' in *Proceedings of the IEEE/CVF winter conference on applications of computer vision workshops*, 2020, pp. 44–52 (pp. 11, 13).

[152] S. Schneider, G. W. Taylor and S. C. Kremer, 'Similarity learning networks for animal individual re-identification: An ecological perspective,' *Mammalian Biology*, vol. 102, no. 3, pp. 899–914, 2022 (p. 13).

[153] S. Schneider, G. W. Taylor and S. C. Kremer, 'Past, present and future approaches using computer vision for animal re-identification from camera trap data,' *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 461–470, 2019 (pp. 11, 14).

[154] D. Schofield et al., ‘Chimpanzee face recognition from videos in the wild using deep learning,’ *Science advances*, vol. 5, no. 9, eaaw0736, 2019 (p. 11).

[155] D. Sculley, ‘Web-scale k-means clustering,’ in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 1177–1178 (p. 89).

[156] Y. Shao, Y. Mei, H. Chu, Z. Chang, Y. He and H. Zhan, ‘Using infrared hog-based pedestrian detection for outdoor autonomous searching uav with embedded system,’ in *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*, SPIE, vol. 10615, 2018, pp. 1143–1147 (p. 35).

[157] D. Shen, G. Wu and H.-I. Suk, ‘Deep learning in medical image analysis,’ *Annual review of biomedical engineering*, vol. 19, no. 1, pp. 221–248, 2017 (p. 18).

[158] R. B. Sherley, T. Burghardt, P. J. Barham, N. Campbell and I. C. Cuthill, ‘Spotting the difference: Towards fully-automated population monitoring of african penguins spheniscus demersus,’ *Endangered Species Research*, vol. 11, no. 2, pp. 101–111, 2010 (pp. 12, 14).

[159] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. d. Carvalho and J. Gama, ‘Data stream clustering: A survey,’ *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, pp. 1–31, 2013 (p. 24).

[160] C. W. Speed, M. G. Meekan and C. J. Bradshaw, ‘Spot the match–wildlife photo-identification using information theory,’ *Frontiers in zoology*, vol. 4, no. 1, p. 2, 2007 (pp. 12, 14).

[161] A. Strehl and J. Ghosh, ‘Cluster ensembles—a knowledge reuse framework for combining multiple partitions,’ in *Journal of Machine Learning Research*, vol. 3, 2002, pp. 583–617 (p. 26).

[162] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, ‘Deepface: Closing the gap to human-level performance in face verification,’ in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708 (p. 19).

[163] T. Tian, J. Zhang, X. Lin, Z. Wei and H. Hakonarson, ‘Model-based deep embedding for constrained clustering analysis of single cell rna-seq data,’ *Nature communications*, vol. 12, no. 1, p. 1873, 2021 (p. 22).

[164] Y. Tian, A. Dehghan and M. Shah, ‘On detection, data association and segmentation for multi-target tracking,’ *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 9, pp. 2146–2160, 2018 (p. 109).

[165] D. Tuia et al., ‘Perspectives in machine learning for wildlife conservation,’ *Nature communications*, vol. 13, no. 1, p. 792, 2022 (p. 11).

[166] J. J. Valletta, C. Torney, M. Kings, A. Thornton and J. Madden, ‘Applications of machine learning in animal behaviour studies,’ *Animal Behaviour*, vol. 124, pp. 203–220, 2017 (p. 11).

[167] J.-A. Vayssade, R. Arquet and M. Bonneau, ‘Automatic activity tracking of goats using drone camera,’ *Computers and Electronics in Agriculture*, vol. 162, pp. 767–772, 2019 (p. 29).

[168] S. Vega-Pons and J. Ruiz-Shulcloper, ‘A survey of clustering ensemble algorithms,’ *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 03, pp. 337–372, 2011 (p. 26).

[169] M. Vidal, N. Wolf, B. Rosenberg, B. P. Harris and A. Mathis, ‘Perspectives on individual animal identification from biology and computer vision,’ *Integrative and comparative biology*, vol. 61, no. 3, pp. 900–916, 2021 (p. 11).

[170] A. G. Villa, A. Salazar and F. Vargas, ‘Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks,’ *Ecological informatics*, vol. 41, pp. 24–32, 2017 (p. 14).

[171] N. X. Vinh, J. Epps and J. Bailey, ‘Information theoretic measures for clusterings comparison: Is a correction for chance necessary?’ In *ICML*, 2009, pp. 1073–1080 (p. 25).

[172] A. S. Voulodimos, C. Z. Patrikakis, A. B. Sideridis, V. A. Ntafis and E. M. Xylouri, ‘A complete farm management system based on animal identification using rfid technology,’ *Computers and electronics in agriculture*, vol. 70, no. 2, pp. 380–388, 2010 (p. 14).

[173] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl et al., ‘Constrained k-means clustering with background knowledge,’ in *Icml*, vol. 1, 2001, pp. 577–584 (pp. 21–23, 76, 121).

[174] G. Wang, M. Song and J.-N. Hwang, ‘Recent advances in embedding methods for multi-object tracking: A survey,’ *arXiv preprint arXiv:2205.10766*, 2022 (pp. 28, 29).

[175] L. Wang et al., ‘Giant panda identification,’ *IEEE Transactions on Image Processing*, vol. 30, pp. 2837–2849, 2021 (pp. 14, 17).

[176] M. Wang, M. L. Larsen, D. Liu, J. F. Winters, J.-L. Rault and T. Norton, ‘Towards re-identification for long-term tracking of group housed pigs,’ *Biosystems Engineering*, vol. 222, pp. 71–81, 2022 (p. 30).

[177] P. Wang, Q. Liu, G. Xu and K. Wang, ‘A three-way clustering method based on ensemble strategy and three-way decision,’ *Information*, vol. 10, no. 2, p. 59, 2019 (p. 26).

[178] Z. Wang, J. Chen and S. C. Hoi, ‘Deep learning for image super-resolution: A survey,’ *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3365–3387, 2020 (p. 20).

[179] Z. Wang, H. Zhao, Y.-L. Li, S. Wang, P. Torr and L. Bertinetto, ‘Do different tracking tasks require different appearance models?’ *Advances in neural information processing systems*, vol. 34, pp. 726–738, 2021 (p. 60).

[180] Z. Wang, L. Zheng, Y. Liu, Y. Li and S. Wang, ‘Towards real-time multi-object tracking,’ in *European conference on computer vision*, Springer, 2020, pp. 107–122 (p. 29).

[181] M. Wasala and T. Kryjak, ‘Real-time hog+ svm based object detection using soc fpga for a uhd video stream,’ in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, IEEE, 2022, pp. 1–6 (p. 35).

[182] G. I. Webb, L. K. Lee, B. Goethals and F. Petitjean, ‘Analyzing concept drift and shift from sample data,’ *Data Mining and Knowledge Discovery*, vol. 32, no. 5, pp. 1179–1199, 2018 (p. 44).

[183] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen and F. Petitjean, ‘Characterising concept drift,’ *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, 2016 (p. 44).

[184] S. Wei, Z. Li and C. Zhang, ‘Combined constraint-based with metric-based in semi-supervised clustering ensemble,’ *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 7, pp. 1085–1100, 2018 (p. 27).

[185] H. J. Weideman et al., ‘Integral curvature representation and matching algorithms for identification of dolphins and whales,’ in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2831–2839 (p. 16).

[186] M. Willi et al., ‘Identifying animal species in camera trap images using deep learning and citizen science,’ *Methods in Ecology and Evolution*, vol. 10, no. 1, pp. 80–91, 2019 (p. 14).

[187] F. Williams, L. I. Kuncheva, J. J. Rodríguez and S. L. Hennessey, 'Combination of object tracking and object detection for animal recognition,' in *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*, IEEE, 2022, pp. 1–6 (pp. 63–66).

[188] F. J. Williams, S. L. Hennessey and L. I. Kuncheva, 'Animal re-identification in video through track clustering,' *Pattern Analysis and Applications*, vol. 28, no. 3, p. 125, 2025 (p. 113).

[189] F. J. Williams, S. L. Hennessey, L. I. Kuncheva, J. F. Diez-Pastor and J. J. Rodriguez, 'A constrained cluster ensemble using hierarchical clustering methods,' in *2024 IEEE 12th International Conference on Intelligent Systems (IS)*, IEEE, 2024, pp. 1–6 (pp. 122, 127–129).

[190] C. L. Witham, 'Automated face recognition of rhesus macaques,' *Journal of neuroscience methods*, vol. 300, pp. 157–165, 2018 (p. 17).

[191] D. H. Wolpert and W. G. Macready, 'No free lunch theorems for optimization,' *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997 (p. 21).

[192] B. Wu, Y. Zhang, B.-G. Hu and Q. Ji, 'Constrained clustering and its application to face clustering in videos,' in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 3507–3514 (p. 22).

[193] Z. Wu, A. Thangali, S. Sclaroff and M. Betke, 'Coupling detection and data association for multiple object tracking,' in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 1948–1955 (p. 109).

[194] K. Xia, X. Gu and Y. Zhang, 'Oriented grouping-constrained spectral clustering for medical imaging segmentation,' *Multimedia Systems*, vol. 26, no. 1, pp. 27–36, 2020 (p. 22).

[195] G. Xian, 'Cyber intrusion prevention for large-scale semi-supervised deep learning based on local and non-local regularization,' *IEEE Access*, vol. 8, pp. 55 526–55 539, 2020 (p. 22).

[196] R. Xu and D. Wunsch, 'Survey of clustering algorithms,' *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005 (pp. 20, 22, 25).

[197] Z. Xu and X. E. Cheng, 'Zebrafish tracking using convolutional neural networks,' *Scientific reports*, vol. 7, no. 1, p. 42 815, 2017 (pp. 29, 30).

[198] R. Yan, J. Zhang, J. Yang and A. G. Hauptmann, 'A discriminative learning framework with pairwise constraints for video object classification,' *IEEE*

transactions on pattern analysis and machine intelligence, vol. 28, no. 4, pp. 578–593, 2006 (p. 22).

[199] F. Yang, T. Li, Q. Zhou and H. Xiao, 'Cluster ensemble selection with constraints,' *Neurocomputing*, vol. 235, pp. 59–70, 2017 (p. 27).

[200] W. Yang, X. Wang, J. Lu, W. Dou and S. Liu, 'Interactive steering of hierarchical clustering,' *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 10, pp. 3953–3967, 2020 (p. 22).

[201] A. Yilmaz, O. Javed and M. Shah, 'Object tracking: A survey,' *Acm computing surveys (CSUR)*, vol. 38, no. 4, 13–es, 2006 (p. 28).

[202] Z. Yu et al., 'Incremental semi-supervised clustering ensemble for high dimensional data clustering,' *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 701–714, 2015 (pp. 26, 27).

[203] M. Zeppelzauer, 'Automated detection of elephants in wildlife video,' *EURASIP journal on image and video processing*, vol. 2013, no. 1, p. 46, 2013 (p. 29).

[204] T. Zhang, Q. Zhao, C. Da, L. Zhou, L. Li and S. Jiancuo, 'Yakreid-103: A benchmark for yak re-identification,' in *2021 IEEE International Joint Conference on Biometrics (IJCB)*, IEEE, 2021, pp. 1–8 (p. 16).

[205] W. Zhang, J. Sun and X. Tang, 'From tiger to panda: Animal head detection,' *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1696–1708, 2010 (p. 12).

[206] K. Zhao, X. Jin, J. Ji, J. Wang, H. Ma and X. Zhu, 'Individual identification of holstein dairy cows based on detecting and matching feature points in body images,' *Biosystems Engineering*, vol. 181, pp. 128–139, 2019 (pp. 11, 12).

[207] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2025 (p. 26).

[208] T. T. Zin, C. N. Phyo, P. Tin, H. Hama and I. Kobayashi, 'Image technology based cow identification system using deep learning,' in *Proceedings of the international multiconference of engineers and computer scientists*, vol. 1, 2018, pp. 236–247 (pp. 11, 13).

[209] A. Zubaroğlu and V. Atalay, 'Data stream clustering: A review,' *Artificial Intelligence Review*, vol. 54, no. 2, pp. 1201–1236, 2021 (p. 24).

[210] A. Zubaroğlu and V. Atalay, 'Online embedding and clustering of evolving data streams,' *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 16, no. 1, pp. 29–44, 2023 (pp. 23, 24).