Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Fine-grained recognition of animals in the wild

Disertation thesis

Mgr. Vojtěch Čermák

Ph.D. programme: Informatics
Supervisor: Ing. Lukáš Neumann, Ph.D.

Prague, January 2025

Supervisor:
Ing. Lukáš Neumann, Ph.D.
Czech Technical University in Prague

Supervisor Specialist:
Mgr. Lukáš Adam, Ph.D.
University of West Bohemia in Pilsen

Declaration

I hereby declare I have written this doctoral thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis. Moreover, I state that this thesis has neither been submitted nor accepted for any other degree.

In Prague, January 2025

.....
Mgr. Vojtěch Čermák

Abstract

Identifying individual animals is crucial for wildlife research and conservation, enabling tracking, behavior analysis, and population monitoring. However, traditional animal identification methods, such as tagging and manual inspection, are labor-intensive and impractical for large-scale studies. Automating this process using machine learning has become essential with the increasing availability of large datasets from camera traps and citizen science. This thesis explores animal identification as a computer vision problem, addressing key challenges such as reliance on fine-grained features, class imbalance, environmental variability, and limited data availability. The goal of this work is to improve automated identification methods, making them more accurate, scalable, and robust for real-world applications.

A key contribution is the WildlifeDatasets library, an open-source toolkit for accessing animal identification datasets, alongside WildlifeTools, a suite of methods and tools designed to enhance research replicability and transparency. We also introduce the SeaTurtleID dataset, which includes timestamps and spans a long duration. Using this dataset, we demonstrate the importance of realistic training and evaluation splits to prevent data leakage and ensure unbiased evaluation. Additionally, we present MegaDescriptor, a foundational deep learning model for animal identification that works across multiple species and outperforms existing methods. Finally, we propose WildFusion, a hybrid approach that integrates local feature matching with deep learning through calibrated similarity fusion, improving both accuracy and computational efficiency.

Keywords: animal identification, deep learning, computer vision, fine-grained recognition, wildlife conservation, similarity fusion, open-source tools.

Abstrakt

Identifikace jednotlivých zvířat je klíčová pro výzkum a ochranu divoké přírody, protože umožňuje sledování, analýzu chování a monitorování populací zvířat. Tradiční metody identifikace, jako je manuální značkovaní, jsou však časově náročné a nepraktické pro rozsáhlé studie. S rostoucí dostupností velkých datových souborů z fotopastí a občanské vědy se proto automatizace tohoto procesu pomocí strojového učení stala nezbytnou. Tato práce zkoumá identifikaci zvířat jako problém počítačového vidění a řeší klíčové výzvy, jako je závislost na detailních rysech, nevyváženost tříd, variabilita prostředí a omezená dostupnost dat. Cílem této práce je zlepšit metody automatizované identifikace, aby byly přesnější, škálovatelné a robustní pro reálné aplikace.

Klíčovým přínosem je knihovna WildlifeDatasets, open-source nástroj pro přístup k datasetům pro identifikaci zvířat, spolu s WildlifeTools, sadou metod a nástrojů navržených pro zvýšení replikovatelnosti a transparentnosti výzkumu. Dále představujeme dataset SeaTurtleID, který obsahuje časová razítka a pokrývá dlouhé časové období. Pomocí tohoto datasetu demonstrujeme význam realistického rozdělení trénovacích a testovacích dat, aby se zajistilo nestranné hodnocení. Navíc představujeme MegaDescriptor, základní model hlubokého učení pro identifikaci zvířat, který funguje napříč více druhy a překonává stávající metody. Nakonec navrhujeme WildFusion, hybridní přístup, který kombinuje lokální párování rysů s hlubokým učením prostřednictvím kalibrované fúze podobností, čímž zlepšuje jak přesnost, tak výpočetní efektivitu.

Klíčová slova: identifikace zvířat, hluboké učení, počítačové vidění, jemnozrnné rozpoznávání, ochrana přírody, fúze podobností, open-source nástroje.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my dear wife, Michaela. Her unwavering support, patience, and care for our wonderful children created a safe and nurturing environment that allowed me to fully dedicate myself to my research. Without her, this journey would not have been possible.

I am immensely grateful to my supervisor, Lukáš Neumann, for his patience, guidance, and invaluable feedback. A special thanks to my supervisor specialist, Lukáš Adam, whose enthusiasm, expertise, and curiosity have been instrumental not only in shaping this work but also in sparking the very topic of this thesis, setting me on this path of exploration. I sincerely appreciate Lukáš Picek, whose collaboration and dedication have been fundamental to this research. The outcomes of our fruitful cooperation form the very foundation of this work.

I would also like to extend my gratitude to Prof. Jiří Matas for welcoming me into the Visual Recognition Group and providing me with an inspiring academic environment. Likewise, I am thankful to my colleagues from both the Artificial Intelligence Center and the Visual Recognition Group for their support, encouragement, and valuable discussions along the way.

Contents

Chapter 1

Introduction

1.1 Why animal identification matters

The goal of animal identification is to recognize individual animals within a species based on their unique characteristics, such as markings, patterns, or other distinct features. It is an important tool for wildlife research as it gives scientists a more detailed insight into the lives of individual animals. By identifying individuals, we can track where they go, watch how they act, check on their health and reproduction, and spot any issues like injuries or illnesses that might affect the rest of the population.

Learning about individual animals helps scientists track changes in populations, like birth and death rates, migration patterns, and social structures. It also gives us a better idea of how animals interact with their surroundings. This includes how they interact with humans, whether it is the negative effects of poaching and deforestation or the positive impacts of things like ecological restoration. Overall, it provides valuable feedback for conservation strategies and biodiversity preservation efforts.

One key use of animal identification is counting individual animals, which helps estimate population sizes and growth and is important for understanding whether a population is thriving. It enables informed decisions to promote habitat preservation, such as moving species or breeding programs. Furthermore, individual identification highlights important animals, such as keystone individuals or a dominant breeding pair, allowing for more targeted and effective interventions.

1.2 Automated animal identification

Accurate identification of individual animals is a complex task that requires deep domain knowledge and is often extremely time-consuming due to manual data processing. Traditionally, wildlife researchers have relied on capturing animals and reading identification tags, a process that is costly, time-intensive, logistically challenging, especially for large-scale monitoring, and stressful for the animals, as it often requires physical restraint. Alternatively, animals can be identified through manual visual inspection of images. However, this includes analyzing distinctive markings or patterns of the animals and requires expert knowledge and careful attention to detail. As the scale of wildlife monitoring continues to increase, particularly through the use of camera traps and citizen science, using manual methods on a large scale becomes increasingly unfeasible, which highlights the need for automating the identification process.

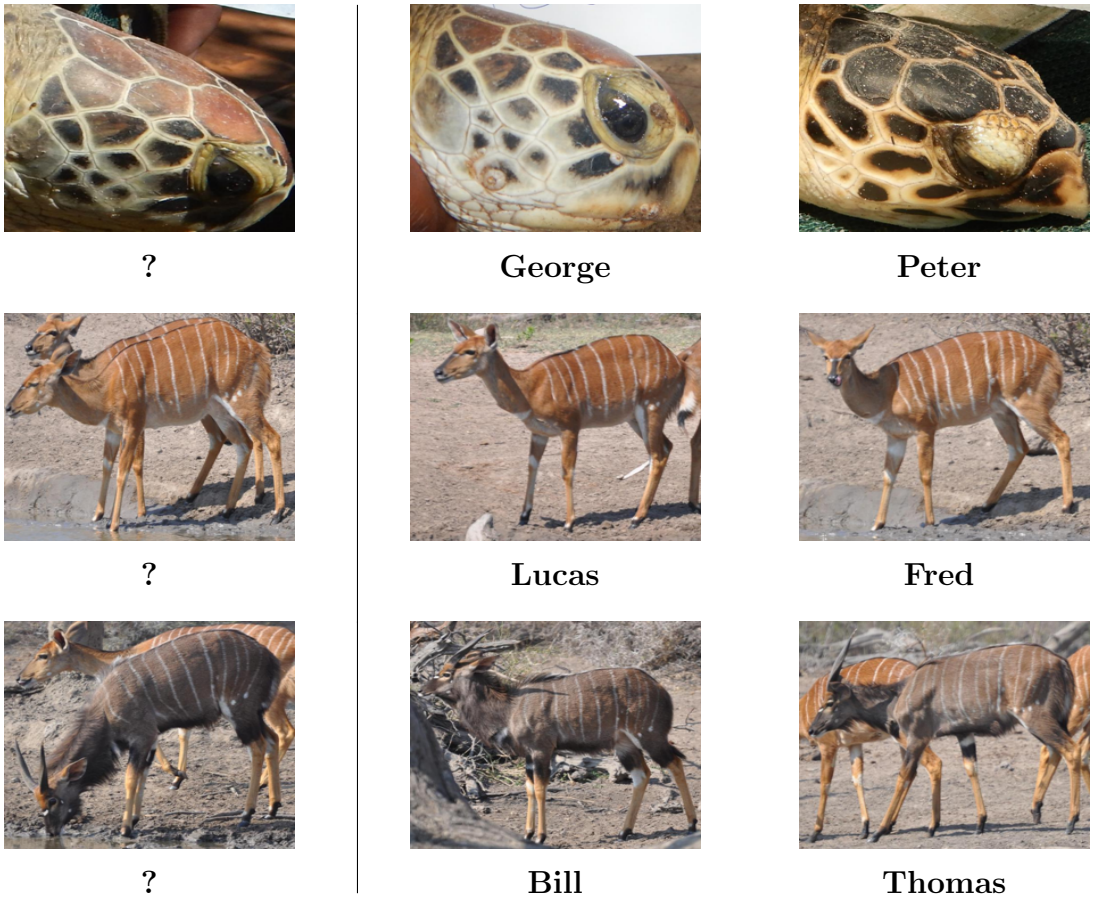


Figure 1.1: Illustration of animal identification problem. The goal is to determine an identity of any animal from an image, given set of known identities. Correct identities in each row are George, Lucas and Bill.

Enabled by recent advances in machine learning and computer vision, automation of the animal identification process provides significant advantages over traditional methods as it is faster and often more accurate. This is particularly beneficial for applications requiring continuous, real-time monitoring, which is nearly impossible to achieve manually. These practical benefits can lead to the development of efficient, data-driven conservation strategies that can rapidly be deployed when needed.

1.3 Animal identification as computer vision problem

From the machine learning perspective, animal identification can be viewed as a classification problem with animal identities as classes. An algorithm is presented with an animal image and is tasked with assigning an identity to the image from a pre-defined set of known animal identities. An illustration of the problem can be seen in Figure 1.1. As opposed to simply predicting the species to which an animal belongs, assigning a known identity to individuals is arguably a more challenging task. For example, it is easy for anyone to recognize pictures of zebras, but differentiating between individual zebras is significantly more difficult for untrained humans as it requires detailed observation and analysis of patterns in zebra stripes. Correspondingly, it is more difficult for machines to utilize fine-grained features that capture the subtle differences between individuals within a species rather than simply learning features that are characteristic of a particular species.

For real-world applications, it is often desirable if the algorithm is able to recognize when it is presented with images of animals that do not belong to any of the known identities and mark it as an unknown identity. This open-set classification setting contrasts with the more commonly used closed-set classification, where it is expected that all classes are in training data. Animal identification can naturally be posed as an open-set classification problem because it is impossible to observe the whole population of animals at once. New individuals may be encountered at any given time, either because they have moved from other regions or have been newly born. These new individuals are not part of the original training set and should be identified as unknown. Open-set classification allows us to collect samples detected as unknown and analyze them or assign them identities for the future. Additionally, open-set classification techniques can often be used for additional downstream tasks, such as clustering unknown identities, which can be applied to help with the labeling to alleviate the manual analysis of collected images.

Animal identification differs from standard person or vehicle identification by emphasizing long-term identification using a few images rather than short-term tracking in video streams from street or security cameras. Despite these differences, the concepts overlap in many datasets we surveyed, such as the datasets from photo traps and farm animals.

1.4 Challenges of animal identification

In animal identification, the assumption that training and test sets are drawn from the same distribution is further violated by variations in observation conditions. These variations include changes in the surrounding environment, lighting conditions, occlusions, and, in the case of underwater environments, additional factors such as water depth and clarity. Additional variations are caused by differences in image capture conditions, such as the type of camera, lenses, or use of a camera flash. All of these factors can affect the appearance of the animals in the images and must be taken into account for accurate identification. Additionally, when dealing with an extensive database of animal individuals that may span multiple years, the model needs to handle changes in the appearance of the animals over time due to aging or injury. All of this puts additional requirements for the model as it must be robust and able to generalize to these variations.

The problem of animal identification presents unique and complex challenges that are highly relevant to advancing computer vision techniques. Issues such as a low number of samples per class, a large number of classes, and significant class imbalances make this task particularly demanding, especially in the context of fine-grained classification and classification in an open-set setting. Many existing benchmarks for those problems rely on toy datasets derived from large artificial image databases like ImageNet. Investigating these challenges in the real-world context of animal identification can push the boundaries of computer vision research, leading to the development of methods that address these problems with meaningful, real-world impact.

1.5 Thesis structure

This work aims to familiarize the reader with automated animal identification, explore its challenges, and guide researchers in avoiding common pitfalls. In addition, we propose several state-of-the-art methods that show promising results in automating this process.

Chapter 2 First, we provide an overview of related work, datasets, methodologies, and tools commonly used in animal identification. This chapter provides a comprehensive overview, including detailed descriptions of evaluation protocols and widely used methods such as deep metric learning, local feature matching, and species-specific approaches.

Chapter 3 This chapter addresses the issue of bias in evaluation when data are randomly split into training and test sets, leading to inflated performance metrics due to data leakage. To explore this challenge, we present **SeaTurtleID**, a novel dataset for animal identification with rich metadata. Using this dataset, we show that using timestamps to create time-aware train-set splits can be used to achieve a more realistic performance evaluation.

Chapter 4 This chapter aims to promote transparency and replicability in animal identification by introducing a toolkit for animal identification. The toolkit consists of **WildlifeDatasets** library for easy access to publicly available datasets and **WildlifeTools** library with state-of-the-art animal identification methods

Chapter 5 In this chapter, we introduce the **MegaDescriptor** line of models, which are designed for individual animal identification across a variety of species. These models significantly outperform existing approaches by jointly modeling multiple identities at once using deep metric learning.

Chapter 6 Finally, we present **WildFusion**, a method that combines local feature matching with deep features using score calibration. We show that WildFusion not only delivers superior performance compared to each method individually but also achieves significant computational reductions in local feature matching.

1.6 Authorship

The thesis is based on the following work:

> L. Adam and V. Cermak, 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*, pp. 7146–7156, 2024

This paper is the basis for the content of Chapter 3 and was done in collaboration with K.Papafitsoros, who collected the **SeaTurtleID** dataset, including annotation and metadata. My contribution to this paper covers methodology and experiments.

> V.Cermak, L.Picek, L.Adam and K. Papafitsoros, "WildlifeDatasets: An open-source toolkit for animal re-identification", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5953–5963, 2024

This paper won the "Best Paper Award" at the Winter Conference on Applications of Computer Vision (WACV 2024). It introduces the **WildlifeDatasets** and **WildlifeTools** libraries as well as the **MegaDescriptor** foundational models. Both Chapter 4 and Chapter 5 build on this work. My contributions include an initial survey of available

datasets and the collection of these datasets into the WildlifeDatasets library, which I co-created with Lukáš Adam. Additionally, I developed the WildlifeTools library, which served as the foundation for implementing the experiments, including the end-to-end training of the final versions of MegaDescriptor.

> V.Cermak, L.Picek, L.Adam and K. Papafitsoros, "WildFusion: Individual Animal Identification with Calibrated Similarity Fusion", in *ECCV Computer Vision for Ecology workshop*, 2024

This paper is the basis for the Chapter 6. Apart from developing and refining the WildFusion idea, my contributions include implementing the method, conducting experiments, and contributing to the writing. Additionally, I integrated and refactored the WildFusion algorithm within the WildlifeTools library. The source for experiments used in this paper significantly extends the WildlifeTools library.

Chapter 2

Related work

In recent years, there has been a surge in research on automatic animal identification, driven by advances in computer vision. However, much of the current research in this area suffers from a lack of unification and connections across different animal species. Many researchers create their own datasets for their specific use case and publish them along with dataset-specific methods, often using inconsistent standards. This makes it difficult for other researchers to use the obtained knowledge for their species of interest. Additionally, this lack of connections often leads to a repetition of poor practices both in problem design and numerical experiments.

While there have been some attempts to survey the field [138, 159, 130, 139], these surveys do not fully cover either datasets or various identification settings. For instance, [139] provides a historical overview but lacks detailed descriptions of methodologies and tasks specific to animal identification. [138] takes a unified approach to identifying multiple animal species but focuses exclusively on deep metric learning methods, applying them to only five animal datasets and one human face dataset. [159] offers valuable insights from a biological perspective, exploring multiple deep learning methods and evaluation settings such as open-set identification. However, their analysis is limited in detail and includes only a small number of datasets. Lastly, [130] focuses on the computer vision side of animal identification, placing it within the larger framework of tracking systems and comparing it to human identification methods. However, their analysis is limited to just 12 datasets.

To address these limitations, we structure this chapter by focusing on four key areas. First, we compile a comprehensive set of animal identification datasets across multiple species, including both private and publicly available resources. Second, we categorize the modeling methodologies into three main approaches: species-specific methods, local feature-based methods, and deep-descriptor-based methods. Third, we examine evaluation methodologies and how they align with different use cases in animal identification. Finally, we add a section on deep metric learning, which has proven to be the most promising approach for animal identification.

Many of the surveyed publications are part of larger pipelines that include tasks such as video sequence analysis, animal detection, cropping, bounding box or segmentation area extraction, pose estimation, and age or sex classification. However, our survey focuses exclusively on the methodology for the identification stage of these pipelines.

Table 2.1: List of animal re-identification datasets. Only papers with datasets published after 2015 are shown. Older papers are shown only for publicly available datasets or in cases, when there are no newer publication on the species.

Species	Publicly available	On request	Private
Bears	[176]	-	[41, 5, 128]
Beetles	-	-	[25, 134]
Birds	[56, 81]	[145]	-
Bumblebees	-	[149]	-
Cats	[30]	[145]	-
Cattle	[9, 7, 8, 61, 93]	-	[14, 175]
Cheetahs	-	-	[79]
Chimpanzees	[57]	[145]	[140]
Deers	-	-	[87]
Dogs	[113, 70]	[145]	[109]
Dolphins	[152]	-	[17, 63, 64, 124, 132, 156]
Elephants	[85]	-	[12, 33, 88]
Ferrets	-	[145]	-
Fish	[22]	-	-
Flies	[137]	-	-
Frogs	-	-	[11, 18, 78]
Giraffes	[42, 107, 123]	-	-
Gorillas	-	-	[23, 21]
Guenons	-	-	[4]
Hedgehogs	-	[145]	-
Hyenas	[153]	-	-
Jaguars	-	-	[118]
Kangaroos	-	-	[28]
Lemurs	-	[45]	-
Leopards	[153]	-	-
Lions	[50]	-	-
Lizards	-	-	[126, 83, 110, 54, 105, 169]
Lobsters	-	-	[58]
Lynxes	-	-	[150]
Macaques	[166]	-	[158]
Manatees	-	-	[91]
Mantas	-	[112]	[151]
Nyalas	[50]	-	-
Ocelots	-	-	[118]
Octopuses	-	-	[75]
Pandas	[161]	[37]	-
Penguins	-	-	[24]
Pigs	-	[145]	[66]
Red pandas	-	-	[69]
Rodents	-	[145]	-
Salamanders	-	-	[55, 114, 131, 134, 52, 106]
Sea lions	-	-	[104]
Sea star	[160]	-	-
Seals	[116]	-	[16, 90]
Sharks	[72]	-	[6, 76]
Sunfish	-	-	[125]
Tigers	[94]	-	-
Turtles	[122, 155, 1]	-	[26, 29, 53]
Water dragons	-	-	[62]
Whales	[13, 34, 77, 133]	-	[84]
Wildebeests	-	-	[111]
Yaks	-	-	[173]
Zebras	[89, 123, 178]	-	-

2.1 Datasets

This section examines approximately 100 papers with datasets, providing a broad and representative overview of the field, though it is not entirely exhaustive. Table 2.1 summarizes the datasets we identified, listing papers that introduce new datasets, categorized by animal types and dataset availability.

Categorizing by animal type is somewhat subjective due to the uneven distribution of datasets. For example, there are numerous datasets for the Artiodactyla order (e.g., whales, dolphins, giraffes, deer, and cattle) but very few for birds. If a dataset spans multiple species, such as HappyWhale [34] and PetFace [145], we included it in all relevant categories. Additionally, since older datasets are often not maintained, we prioritize those published after 2015 when multiple datasets exist for the same animal.

We also categorize datasets on the basis of their availability. Although some datasets are publicly available, some papers only allow access to their datasets upon request for scientific purposes, while others keep their datasets entirely private. Additionally, datasets that are reported as available but are no longer maintained or accessible are categorized as private. All publicly available datasets that we successfully downloaded and analyzed were incorporated into the **WildlifeDatasets** library. We provide brief description of each publicly available dataset in Section 4.2.

2.2 Methods

There are three primary approaches commonly used for animal identification – (i) species-specific methods [12, 163, 64, 79, 5], (ii) local descriptors [132, 8, 53] and (iii) deep descriptors [22, 158, 94, 45, 107]

2.2.1 Species-specific methods

Species-specific methods are tailored to an individual species or groups of closely related species, particularly those with visually distinct patterns. However, these methods typically focus on visual characteristics unique to the target species and are not easily transferable to other species. Moreover, they often involve substantial manual preprocessing steps, such as extracting patches from regions of interest or accurately aligning compared images.

For example, [5] used the Chamfer distance to measure the distance between greyscale patterns in polar bear whiskers. Other examples include computing correlation between aligned patches derived from cheetah spots [79] or similarity between two images based on the count of matching pixels within newt patterns [52]. In [12], they identify individual African elephants by analyzing the tusks or the ear pattern using hand-crafted visual features.

Another line of work is focused on the identification of cetaceans, such as dolphins and whales. For example, finFindR [164] uses integral curvature to analyze nicks and notches along cetacean fin trailing edges, creating a viewpoint-robust representation that enables individual identification through either dynamic programming alignment or feature descriptor matching. In their follow-up work, they introduced CurvRank v2 [163], which they use to extract precise contours of humpback whale flukes and elephant ears using fully connected neural networks, followed by curvature-based matching. In [64],

they identified New Zealand common dolphins by extracting orientation-robust features from dorsal fin pigmentation patterns.

2.2.2 Local-feature based methods

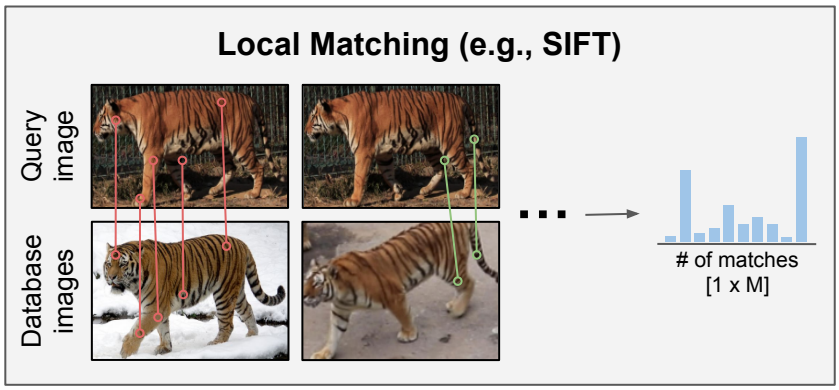


Figure 2.1: **Schema of local-feature based methods.** Similarity between query image and database is determined from matching local keypoints and descriptors. For example, high number of significant SIFT correspondences indicates positive match.

Since species-specific methods often suffer from poor performance and are not transferable to other species, methods extracting local patterns such as SIFT [101], Affine-SIFT [171] or ORB [135] were soon widely used. Local feature methods find unique keypoints and extract their local descriptors for matching. The matching is usually done on a database of known identities, i.e., for each given image sample, an identity with the highest number of descriptor matches is retrieved. For an illustration of the procedure, see Figure 2.1.

The most significant benefit of these methods is their plug-and-play nature, without any need for fine-tuning, which makes them comparable in a zero-shot setting to large foundation models, such as CLIP [129] or DINOv2 [119], etc. The universality of local feature-based methods allows them to be used across a wide variety of animal species. For example, in [53], SIFT-based HotSpotter was applied for the identification of sea turtles, while [132] used SIFT for identifying dolphins. Similarly, [8] employed Affine-SIFT for the identification of cattle.

Recently, the focus has moved to local features extracted by deep networks such as ALIKED [174], DISK [157] or SuperPoint [49]. The classical matching of local descriptors could be simply replaced by deep methods such as LightGlue [96], Superglue[136], and LoFTR [147] that allow both extracting and matching of the local features. These matching methods return potential matches and their confidence scores. They require manual thresholding to determine which features are matched. In animal identification, deep local features are slowly coming into focus; for example, [125] used a combination of the SuperPoint features with the SuperGlue matching.

Even though approaches based on local descriptors exhibit limitations in scaling efficiently to larger datasets and their performance, all available software products, e.g., WildID [15], HotSpotter [44], and I^{3}S [19], are based on local-feature-based methods. Naturally, even with such limitations, those systems are popular among ecological researchers without a comprehensive technical background and find a wide range of applications, most

likely due to their intuitive graphical user interfaces (GUIs).

2.2.3 Deep learning methods

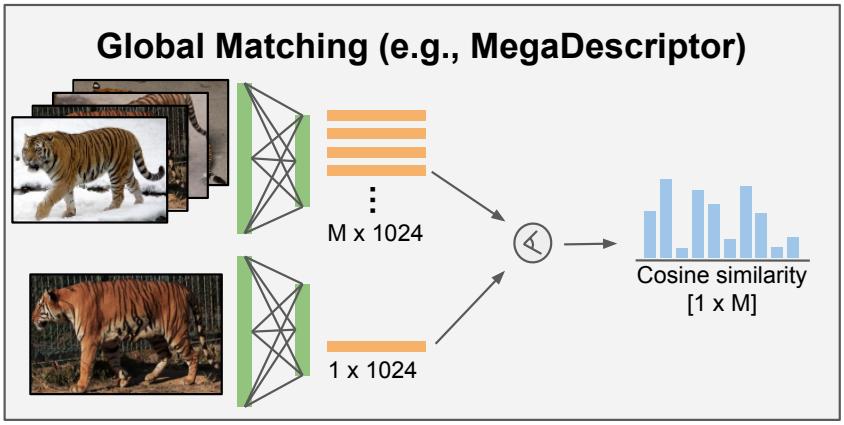


Figure 2.2: **Schema of deep embedding methods.** Similarity between query image and database is determined from similarity between global embeddings. For example, high cosine similarity between MegaDescriptor embeddings indicates a positive match.

One of the earliest applications of deep learning methods in animal identification is described in [29], where a fully connected network was used to determine whether two turtle images represent the same individual. Another use case consists of extracting embeddings from a neural network and feeding them to an SVM classifier [109, 41, 85]. This approach has low computational demands, but the network backbone cannot be fine-tuned. Another simple approach involves fine-tuning a pre-trained neural network [56, 137]. Both those approaches usually require a fixed number of classes (individuals), which is not realistic.

For this reason, deep metric learning methods (e.g., ArcFace [31], Siamese networks [78], and Triplet loss [50, 107]) became popular. Instead of classifying images into a predetermined set of classes, they measure differences between images and are, therefore, able to generalize into new individuals. Central to these methods is learning how to represent images as global vectors by optimizing a deep neural network. The resulting deep embeddings, vectors with a fixed number of dimensions, are then matched against an identity database to identify individuals. For an illustration of the matching using deep embeddings, refer to Figure 2.2.

Applying deep learning to animal identification bears similarities with human or vehicle re-identification. Therefore, similar methods can be easily repurposed. However, it is important to note that deep learning requires fine-tuning models on the specific target domain, i.e., species, which makes the model's performance dependent on a species it was fine-tuned for.

Another approach is to use publicly available large-scale foundational models pre-trained on large datasets such as CLIP [129], BioCLIP [146] and DINOv2 [119]. Since these models are primarily designed for general computer vision tasks, they are not adapted for the nuances of animal identification, which heavily relies on fine-grained patterns. We address this by introducing the foundational MegaDescriptor model in

Chapter 5, the Swin-based model for animal identification that was trained on 30 public datasets (collected using WildlifeDatasets) using ArcFace loss [47]. A similar approach for the animal identification foundation model is MiewID [120], which used an EfficientNet[148] backbone trained on a collection of more than 60 mostly private datasets.

2.3 Evaluation protocols

Animal identification does not have a standardized definition in literature. It generally involves assigning the identity of an individual animal based on an image and is considered a supervised learning problem because labeled data is available for training a model. When evaluating the performance of a machine learning algorithm, it is necessary to choose an evaluation protocol that defines how to split the dataset into a training set used for training the model and a test set used for evaluating its performance. Several considerations must be considered when designing the evaluation protocol to ensure that the evaluation scenarios are realistic and accurately reflect real-world applications.

One key consideration is whether the evaluation follows a closed-set or open-set scenario. In a closed-set scenario, every image presented during inference corresponds to a known individual, and all identities are predefined. The term "closed-set" reflects the assumption that the population is fully known, meaning no new individuals appear during inference. The goal in this setting is to assign one of these known identities to each test sample.

In contrast, an open-set scenario allows for new, previously unseen individuals during inference. This setting is more realistic, as new animals are frequently encountered in practical applications. The objective here is twofold: recognizing known identities while also detecting and distinguishing new individuals.

Animal identification can also be viewed from a methodological perspective. From a machine learning standpoint, this task is often framed as a classification problem, where each test image is assigned to a predefined identity class. A more flexible alternative is image retrieval, in which a query image is compared against a database of known individuals to find the most similar image. Both classification and image retrieval can be applied in either a closed-set or open-set setting, where new individuals may appear in the test set or query set, respectively.

Finally, the temporal aspect introduces another dimension to evaluation scenarios. Identification over time considers factors such as when an observation was made and how an individuals appearance may change over time. This perspective is particularly relevant in longitudinal studies, where animals are tracked across different time periods.

2.3.1 Identification as classification

Identification can be performed using standard classification methods such as a classifier with softmax activation and cross-entropy (CE) loss or support vector machines (SVM) [43] using visual features. To evaluate a classification model for the closed-set task, the dataset is typically split randomly into training and test subsets with the same individuals in both training and test sets. Standard classification accuracy is the most commonly used evaluation metric in this context.

Incorporating new identities requires retraining the model, which can be time-consuming and resource-intensive. Therefore, the classification approach is well-suited to controlled

and stable observation environments, such as zoos, livestock farms, or private game reserves, where conditions remain relatively consistent. In contrast, its use for wild animal populations is limited, as the assumption of a fixed, fully observed population is unrealistic in such dynamic settings. Despite these limitations, the classification approach is widely used in the literature due to its simplicity and direct connection to other classification problems in the computer vision domain (e.g., [92, 86, 46]) as it does not require specialized modifications to commonly used classifiers.

Convolution networks trained with cross-entropy loss were used in [56, 175, 140] for the identification of birds, cattle, and chimpanzees, respectively. A less computationally intensive approach that involved fine-tuning only the last layer of an ImageNet-pretrained network was applied in [158, 37] for identifying macaques and pandas. Similarly, [23, 33, 85, 109] used features extracted from deep neural networks combined with SVM classifiers to identify gorillas, cats (e.g., jaguars and tigers), and elephants. In [66], they compared pig identification methods, showing that the convolutional network trained with cross-entropy loss outperformed the SVM classifier applied to extracted features, highlighting the advantages of end-to-end training. In [4], they used species-specific features in combination with an LDA classifier for guenon identification. A similar approach with the LDA classifier and with dolphin-specific features was used in [124].

2.3.2 Identification as image retrieval

An alternative approach to animal identification is using image retrieval, which leverages pairwise similarity comparisons between images. The process starts by creating a database of images with known identities. During inference, a query image with an unknown identity is compared to the database to find the most similar matches. This approach is closely related to classification since an image can be assigned a label based on its closest matches, effectively making it a nearest-neighbor classifier. However, using image retrieval for animal identification offers greater flexibility.

A key advantage of this method is that pairwise matching is guided by prior knowledge, such as feature representations learned during previous training. As a result, the model can be trained on entirely different datasets with disjoint sets of identities or, in some cases, may not require training at all. In addition, expanding the database with new identities is straightforward and does not require retraining the model.

The image retrieval approach is also naturally suited for ranking tasks, allowing predictions of top-k identities. This makes it effective in scenarios where multiple possible matches need to be evaluated, a feature particularly useful for biologists conducting manual verification of results. Similar methods are widely used in face recognition benchmarks [80, 74]. However, in animal identification, the specifics of database and query set construction often depend on the application. For evaluation, metrics common in image retrieval, such as top-1 or top-k accuracy, are typically used to measure performance.

Alternatively, pair verification can be used to evaluate the quality of algorithms for pairwise matching. The pair verification task involves comparing two images to determine whether they depict the same individual. In this case, binary classification metrics, such as the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), are used to assess performance.

Image retrieval using metric learning

In metric learning-based image retrieval, pairwise similarity is calculated using embeddings extracted by deep neural networks. These embeddings capture high-level feature representations, which can be used for effective calculation of similarity. The neural network is first trained, and the learned feature representations are used during inference to compute distances between images. Training can be conducted using either a subset of the analyzed dataset or using an entirely different dataset. For evaluation, image retrieval is performed between a database of images and a query set, typically both derived from the test set. However, variations in evaluation settings are commonly explored in the literature to fit different use cases.

Metric learning is commonly used in animal identification literature. For example, [94] and [17] evaluated various metric learning approaches for the identification of tigers and dolphins, respectively. The models were trained in a training set and evaluated in a test set with a disjoint set of identities. Each test sample was used as a query, and the remaining samples formed the database. The same approach was used for the identification of species such as nyalas, lions, and chimpanzees in [50], but they split the dataset into training and test sets randomly.

A slightly different evaluation strategy of metric learning algorithms was used in [112] for manta rays and [45] for lemurs and other primates. In particular, [45] expanded the standard disjoint identity setting to an open-set variant, introducing queries with identities absent from both the database and training set to assess the model's ability to handle unseen identities.

Both studies also incorporated a verification task that evaluated the performance of the model in positive and negative pairs within the test set. The same pair verification setting was used in the evaluation of models for the identification of toads and whales in [78]. Pair verification was also used in [41], where they used it to evaluate the quality of metric learning embeddings. Then, they used the embeddings as features for an SVM classifier to identify brown bears.

In cattle identification, [14] used metric learning methods like SphereFace [97] and EigenFace [154] with three dataset splits. It was split into a training set with disjoint identities from the database and a query set. Each query sample was matched against the database to return k nearest matches.

Another approach was adopted by [69] to evaluate the identification of red pandas. Here, the training and test sets had disjoint identities, but the test set was split so that 50% of the images were used as queries, the remaining forming the database. In addition, all training samples were included in the database as distractors, providing a more challenging retrieval scenario. Finally, for giraffe identification, [107] compared metric learning with SIFT-based matching. Their evaluation used a database with five images per identity from the training set, while test set images were used as queries.

Image retrieval without training

Calculating pairwise similarity in image retrieval does not necessarily require training. Many methods rely on classical computer vision techniques, including species-specific approaches and methods that match images using local keypoints and descriptors. Popular tools in this category include Hotspotter [44], WildID [15], and I^{3}S [19]. Since these methods are not based on machine learning and do not require training, they effectively operate in a zero-shot regime.

Biologists widely use these tools for population monitoring studies. A common use case involves biologists maintaining their database of known, identified animal images. New images are then compared to this database under manual supervision and, if validated, are subsequently added to the database.

I^3S software has been extensively used for identification tasks across various species, including whale sharks [72], beetles [25], lizards [126], perenties [110], and water dragons [62].

WildID [15] has demonstrated versatility, being employed for identifying beetles and salamanders [134], newts [106], and salamanders in another study [55]. It has also been compared with other tools for seal identification [90, 105].

Hotspotter [44] has been applied to identify turtles [53], jaguars, and ocelots [118], as well as being compared with custom local feature matching methods for seal identification [116].

The APHIS tool was introduced in [114] for amphibian identification, with experiments conducted on lizards. In [11], APHIS was evaluated alongside WildID for frog identification. Proprietary matching software has been used for wildebeests, sharks, and various lizards [111]. FinFindR [164] and CurvRank have been developed for marine mammal identification [156]. The SEEK tool focuses on elephant ear curvature matching [88, 12].

Several studies have evaluated and compared identification tools. For instance, [90, 105] assessed WildID and I^3S for seals, while [11] and [118] compared multiple tools like WildID, AMPHIS, and Hotspotter across different species.

2.3.3 Open-set setting

A more realistic scenario for animal identification involves the open-set setting, where we assume that, during inference, there are individuals who are unknown and were not previously seen. These individuals could include newly born animals or animals that avoid paths where camera traps are located. The goal is to design an algorithm that either assigns one of the known identities to a sample or predicts that the image depicts a new, unknown individual. This prediction can be in the form of a novelty score, which quantifies how likely a sample belongs to the unknown class. Animals detected as new can then be manually examined by researchers and labeled with an identity.

The closed-set setting scenario can be converted into the open-set scenario by adding new classes during inference. In the classification approach, a subset of individuals is selected, and all their samples are assigned exclusively to the test set. The remaining individuals have their samples randomly split between the training and test sets, following the standard closed-set methodology. Similarly, the open-set setting can be simulated for the image retrieval approach by reserving some individuals only in the query set and excluding them from the database entirely. The extent to which the population is partially observed can be controlled by adjusting the number of individuals that are exclusively used during inference.

Open-set classification is more complex than closed-set classification, as it requires both detecting new identities and classifying known ones. One approach is to use closed-set metrics such as accuracy and add "unknown" as a new class. Alternatively, performance can be evaluated with two separate metrics: one for detecting unknown classes and another for classifying known classes. Detecting unknowns can be treated as a binary classification task, and standard metrics such as AUC-ROC can be used to evaluate performance across all novelty score thresholds. Classification performance is measured

only on known-class samples, simulating a perfect detector for unknowns.

Several studies have explored animal identification in scenarios involving new or unknown individuals. For instance, [56] proposed detecting new bird individuals by applying a threshold on the entropy of predicted probabilities. [107] trained a metric learning model for giraffe identification and evaluated its performance on a test set that included both known identities from the training set and new, unseen identities. Their evaluation involved constructing a database with five images per identity from the training set, using all images from the test set as queries. A similar evaluation was done on lemurs [45] and newts [52], which assessed the ability of algorithms to detect unknown individuals by adding new identities in the query set. In cattle identification, [9] adopted an open-set evaluation approach by withholding certain classes from the training set.

2.3.4 Time-aware setting

The time aspect is important in animal identification because it provides a way to indicate different encounters and the corresponding factors that influence the identification of animals. Factors such as changes in the surrounding environment, image capture conditions, and changes to the individual animals can all vary from encounter to encounter. The most efficient way to indicate these differences in a dataset is by including the capture time, or timestamps, in the metadata. Without timestamps, datasets can only be split into training and test sets randomly, which leads to an overestimation of the generalization ability of the methods to recognize individuals in future encounters. Time-aware splits, on the other hand, allow for a more realistic case when new factors are encountered in the future. Figure 2.3 illustrates train-test splits based on time. Black crosses represent images, and round shapes group together images of the same individual. The horizontal axis shows how images were observed over time. Time-aware splits (second and third row) are done with respect to the horizontal axis, as can be seen from the parts filled with light grey and dark grey, which correspond to the training and test set, respectively.

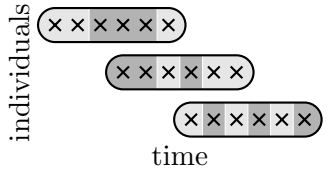
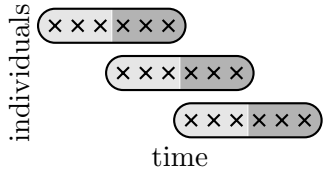
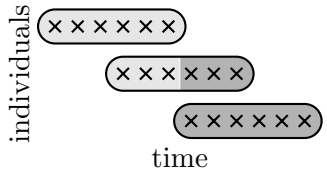
Split name	Schema	Setting	Description
Time-unaware		Closed-set	The split into train (light gray) and test set (dark gray) is random (not based on time).
Time-proportion		Closed-set	For each individual (vertical axis), the first half of samples go to the train set (light gray), the second half to the test set (dark gray).
Time-cutoff		Open-set	Samples before the time threshold go to the train set (light gray), those after to the test set (dark gray).

Figure 2.3: Illustration of various time-aware evaluation scenarios.

Time-unaware split

The time-unaware split (first row in Figure 2.3) corresponds to the standard approach in supervised machine learning to evaluate classification models. In this approach, we randomly split the dataset into training and test sets without considering any timestamps. In the case of extreme class imbalance, it is a good practice to perform the split in a stratified way such that the ratio of training and test set samples is constant for all individuals. Time-unaware split leads to a closed-set classification problem if stratified sampling is used. Most of the surveyed literature adopts this approach, as time-based splits are only sporadically considered.

Time-proportion split

In the time-proportion split, we split the dataset based on timestamps such that the proportion of samples in training and test sets remains approximately the same for all individuals. For each individual, we sort the images by time in chronological order. Given a train-test proportion, we assign the corresponding number of images to the training and test sets while respecting the chronological order so that images of an individual taken on the same day belong to only one of these sets. An illustration of this split variant can be seen in the second row of Figure 2.3, which corresponds to 0.5 train-test proportion. Time-proportion split always leads to the closed-set classification problem. This evaluation protocol was introduced in [2]. Similar evaluation was used in [128], were they used classical computer vision techniques to match images of polar bears collected across different encounters. They created sets of images, each representing a single individual from one encounter, and then performed pairwise comparisons between sets sampled from different encounters.

Time-cutoff split

A more realistic scenario is the time-cutoff split (third row of Figure 2.3). It consists of dividing the dataset by selecting a cutoff point, for example, some date, after which all samples are in the test set. In this way, we respect the temporal order in which the data were collected. Time-cutoff can be used for modeling scenarios where it is important to respect the data collection order. It can include repeated collections of the data over time, which leads to a sequence of cutoff splits.

As an example, we consider a situation where a researcher is tasked with long-term observation and mapping of a wildlife population. In each time step, new images are obtained, and the goal is to assign them identities from a set of all known identities up to the date or label them as unknown. In other words, this means making predictions in each time step based on information from the training set consisting of samples collected up to the date. In the next step, the training set with known identities is expanded by adding all images from the test set of the previous step to the training set. This process typically requires regular manual re-labeling of new animals by the curators at the end of each time step. A time-cutoff typically leads to an open-set classification problem.

The time-cutoff setting has mainly been studied in biology-focused research, often alongside specific software tools. However, it is rarely discussed in papers on identification method development, highlighting a gap in the machine learning perspective, even though it is highly relevant to biological studies.

For example, [140] investigated chimpanzee identification using a closed-set classification, where frames from older videos were used for training and from newer ones for testing. Similarly, [156] applied a time-cutoff setting for dolphin identification in an image retrieval setup, using software like CurvRank, with database images collected in years prior to query images. Sea turtles were analyzed by [53], who employed Hotspotter software in a time-aware setting, where database images were collected in earlier years than query images. For seals, [90] used software I^3S and WildID such that query and database images were taken on different days.

Lizards were studied in [126], which utilized the I^3S [19] software to perform both within-year and between-year splits for query and database images. In the case of newts, [106] employed WildID for identification, with database and query images drawn from different years. For salamanders, [114] evaluated image matching by designating photos from a single day as the initial database and using the remaining images as the query set. Finally, [55] utilized WildID software to analyze how individual salamanders change over time. In one of their experiments, they examined whether individuals remained identifiable across different time points.

2.4 Metric learning

Metric learning is a machine learning technique that aims to learn to measure similarity between data points. This is usually done by learning a function f_θ that maps input samples x from some input space $\mathcal{X}$ to some feature space. The representation in the feature space should group similar samples together. The feature extractor f_θ, often represented by a neural network, is parameterized by unknown parameters θ, which are determined in a process called training.

Features obtained from metric learning algorithms can be useful for a variety of downstream tasks, including classification, clustering and few-shot learning. One common application of metric learning is in the field of face recognition, where it is used for tasks such as face verification and identification. In face verification, the goal is to determine whether two given faces belong to the same individual. This is typically done by using the learned distance function produced by the metric learning algorithm to measure the similarity between the two faces. If the distance between the faces is below a certain threshold, it is considered a match, indicating that the faces belong to the same individual. In face identification, the goal is to find the closest match in a database using feature similarity from a metric learning algorithm. A large database of faces is typically used to train a metric learning algorithm for face recognition [74, 117, 80]. This database is often sourced from the internet and contains a set of faces representing a wide range of identities. On the other hand, the performance of a face recognition algorithm is typically evaluated on a disjoint set of faces or identities rather than on the same database used for training. This is because, in real-life applications, it is important for the algorithm to be able to recognize any face identity, not just those it has seen during training.

In the context of open-set classification and anomaly detection, metric learning can be used to learn features that effectively separate known classes from unknown or anomalous classes, as those should be significantly distant in the feature space from the known classes.

2.4.1 Direct deep metric learning

During training, we sample input pairs of (x_i, x_j) and pass them through two copies of f_θ with shared parameters to extract embedding vectors $(f_\theta(x_i), f_\theta(x_j))$ which are further used for calculating the distance. Direct deep metric learning operates directly in the feature space, where it measures the Euclidean distance between features by

$$D_\theta(x_i, x_j) = ||f_\theta(x_i) - f_\theta(x_j)||_2. \tag{2.1}$$

In the supervised setting, we want samples with the same label to have embedding vectors $f_\theta(x_i)$ close to each other and samples with different labels far from each other. The former amounts to small D_θ, while the latter to large D_θ.

Constrastive loss

The main idea behind contrastive loss [40, 65] is to pull together samples with the same label and push apart samples with different labels. The parameters θ of distance $D_\theta(x_i, x_j)$ in (2.1) can be learned using the contrastive loss with siamese architecture [20]. In addition to the distance, contrastive loss needs binary label $Y \in [0, 1]$ as an input, which corresponds to the degree of similarity between the samples $(x_i\ x_j)$. We set $Y = 1$ if the label of x_i is the same as the label of x_j and $Y = 0$ otherwise. We define the contrastive loss by

$$\mathcal{L}_{\text{contrast}}(x_i, x_j, Y) = Y D_\theta(x_i, x_j)^2 + (1 - Y)(\max[0, m - D_\theta(x_i, x_j)])^2. \tag{2.2}$$

Its first part minimizes the distance $D_\theta(x_i, x_j)$ between points with the same label which pulls them together, while its second part is responsible for pushing differently labeled points apart. The margin parameter m defines the minimum distance that must be maintained between points with different labels, beyond which no penalty is applied. In other words, two points with different labels will be pushed apart only if the distance $D_\theta(x_i, x_j)$ between them is smaller than m. If the distance between them is larger than m, they are considered to be apart enough and the loss becomes zero.

In practical implementations, sampling of the (x_i, x_j) pairs is done using some mining strategy that describes how to create both negative and positive pairs.

Triplet loss

In the triplet loss [144], we are given triplet of (x_a, x_p, x_n) such that the anchor sample x_a have the same label as the positive sample x_p and different label than the negative sample x_n. The triplet loss

$$\mathcal{L}_{\text{triplet}}(x_a, x_p, x_n) = \max[0, D_\theta(x_a, x_p)^2 - D_\theta(x_a, x_n)^2 + m] \tag{2.3}$$

learns a representation that minimizes the distance between x_a and x_p and maximizes the distance between x_a and x_n. The former distance squared should be smaller than the latter distance squared, at least by a margin m.

The triplet loss is particularly sensitive to the triplet selection. For example, consider the situation, where the negative sample is substantially distant from the anchor. Then, the term $D_\theta(x_a, x_n)$ is large and outweights any effect of the positive sample and the loss is zero. In that case, the model learns nothing from the triplet. When triplets are selected randomly, this situation occurs more often as the training converges. Therefore, suitable hard negative mining strategy for triple selection can be used to boost performance of the algorithm [71].

2.4.2 Classification-based deep metric learning

Another approach to metric learning is based on extending standard classification models to better perform metric learning tasks. Usually, models in this category consist of a feature extractor f_θ, also called backbone, followed by classification head parameterized by matrix W, where vector W_j is its j^{th} row.

Softmax loss

We can split the standard classification model in the penultimate layer to feature extractor f_θ and the classification head in the form of a dense layer with softmax activation. The output of the classification head is directly used in calculating the crossentropy loss. This leads to the softmax loss

$$\mathcal{L}_{\text{softmax}}(x_i, y_i) = -\log \frac{e^{W_{y_i}^T f_\theta(x_i) + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T f_\theta(x_i) + b_j}}. \tag{2.4}$$

The total loss function is an average over all input samples.

This setup can be used for the same downstream tasks as any other metric learning algorithm by using the embedding vectors from the feature extractor backbone.

ArcFace

The softmax loss is only concerned about learning good classification features and does not explicitly put any constrains on the feature embeddings. In other words, it does not explicitly enforce that samples with the same label are close to each other and far from differently labeled samples. Although these features can still be used for the downstream tasks, it can pose problems for domains like facial recognition where the goal is to learn features with high inter-class diversity to recognize wide variety of faces and with high intra-class compactness to overcome variations in the faces.

ArcFace [47] was designed to overcome those problems. Starting from (2.4), it sets bias term $b_j = 0$ and continues with $W_j^T f_\theta(x_i) = \|W_j\| \|f_\theta(x_i)\| \cos(\alpha_j)$, where α_j is the angle between weight W_j and feature vector $f_\theta(x_i)$. Additionally, it normalizes the weight vectors such that $\|W_j\| = 1$ and the embedding vectors such that $\|f_\theta(x_i)\| = s$, where s is a scaling hyperparametrer. This puts both the weight and the embedding vectors on the hyper-sphere with radius s. An additive angular margin m is then added between $f_\theta(x_i)$ and W_j to improve the discriminative power. This leads to the ArcFace loss

$$\mathcal{L}_{\text{arcface}}(x_i, y_i) = -\log \frac{e^{s\cos(\alpha_{y_i} + m)}}{e^{s\cos(\alpha_{y_i} + m)} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos(\alpha_j)}}. \tag{2.5}$$

The ArcFace loss is minimal when $\cos(\alpha_{y_i})$ is large, which means that the the angle between W_{y_i} and $f_\theta(x_i)$ is small. In other words, target vectors W_j can be interpreted as cluster centers for each class. This interpretation of W_j is different to the softmax loss, where it defines the decision boundary.

Sub-center ArcFace

The performance of ArcFace can suffer in performance because it forces all embeddings of one class to be part of one cluster. This can be particularly problematic for datasets that naturally have large inter-class variance or in presence of outliers.

For example, in the case of turtle identification, each turtle can be uniquely identified by scales from each side of head, which are unique for each of the side. In this case, it makes sense to create two cluster centers, one for each side of head. To mitigate those problems [48] introduced sub-center ArcFace, which extends the original ArcFace to have multiple class centers. The key difference is that they added additional dimension to the normalized weights $W_{j,k}$ where $k = 1, \ldots, K$ and K is the number of sub-centers. The sub-center ArcFace loss

$$\mathcal{L}_{\text{sc-arcface}}(x_i, y_i) = -\log \frac{e^{s\cos(\tilde{\alpha}_{y_i,i}+m)}}{e^{s\cos(\tilde{\alpha}_{y_i,i}+m)} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos(\tilde{\alpha}_{j,i})}} \tag{2.6}$$

resembles the ArcFace loss (2.5) but the angles α_j are replaced by the modified angles

$$\tilde{\alpha}_{j,i} = \arccos(\max_k W_{j,k}^T f_\theta(x_i)) \tag{2.7}$$

which measure the angle between $f_\theta(x_i)$ and the closest sub-center defined by the rows of the W matrix.

Having multiple cluster center for each class helps in situation with noisy, mislabeled data, where the majority of normal data will be covered by one or few dominant clusters and outliers will be placed in one of the remaining sub-centers. Including additional sub-centers can help to cover wider variety of outliers.

Chapter 3

Time-aware identification

In this chapter, we demonstrate that standard evaluation protocols using random dataset splitting leads to significant overestimation bias, that can be easily remedied by evaluation using time-aware splits. For our experiments, we introduce SeaTurtleID2022, a novel dataset for animal identification featuring diverse annotations, including identities, encounter timestamps, segmentation masks, bounding boxes, and body part orientations.

We provide empirical evidence of overestimation bias from time-unaware splits, which significantly inflate performance compared to time-aware splits. Therefore, we recommend using time-aware splits for animal identification evaluation and urge dataset collectors to include timestamp in their metadata. Building on our findings, we present and evaluate an end-to-end system for reliable animal identification in the wild, designed to be easily adaptable to other species.

For this dataset, we provide baseline identification performance using both metric learning and SIFT descriptor matching approaches. Additionally, we report baseline results for body-part segmentation using well-known instance segmentation methods. Our results suggests that higher accuracy on SeaTurtleID2022 can be achieved on cropped head images. However, full image identification performed poorly without body part detection, emphasizing its importance in sea turtle identification.

3.1 Motivation

The quality of datasets influences the objectivity of the method evaluation. Therefore, the dataset and its splitting should mimic a realistic scenario, i.e., the images in the *query* and *database* sets should not originate from the same *encounters* (burst mode in camera traps, consecutive video frames, multiple photographs taken during an encounter). Other *factors*, e.g., different locations, image capture conditions, and images that reflect changes in animal appearances over time, are also vital. For reference, see Figure 3.1, shows sample images from the SeaTurtleID2022 dataset, highlighting the variety of photographs (poses, orientations, backgrounds, etc.).

Typically, images produced during one *encounter* share the same factors as the encounter lasts for a short period. The most efficient way to indicate different encounters and factors in a dataset is by including the capture date and time in metadata, i.e., *timestamps*. Without knowing the time of the observation, datasets are often split into database and query sets exclusively randomly. Therefore, images in training and test sets often originate from the same encounter/observation, representing unwanted training-to-test data leakage. This might result in overfitting to factors of a particular encounter

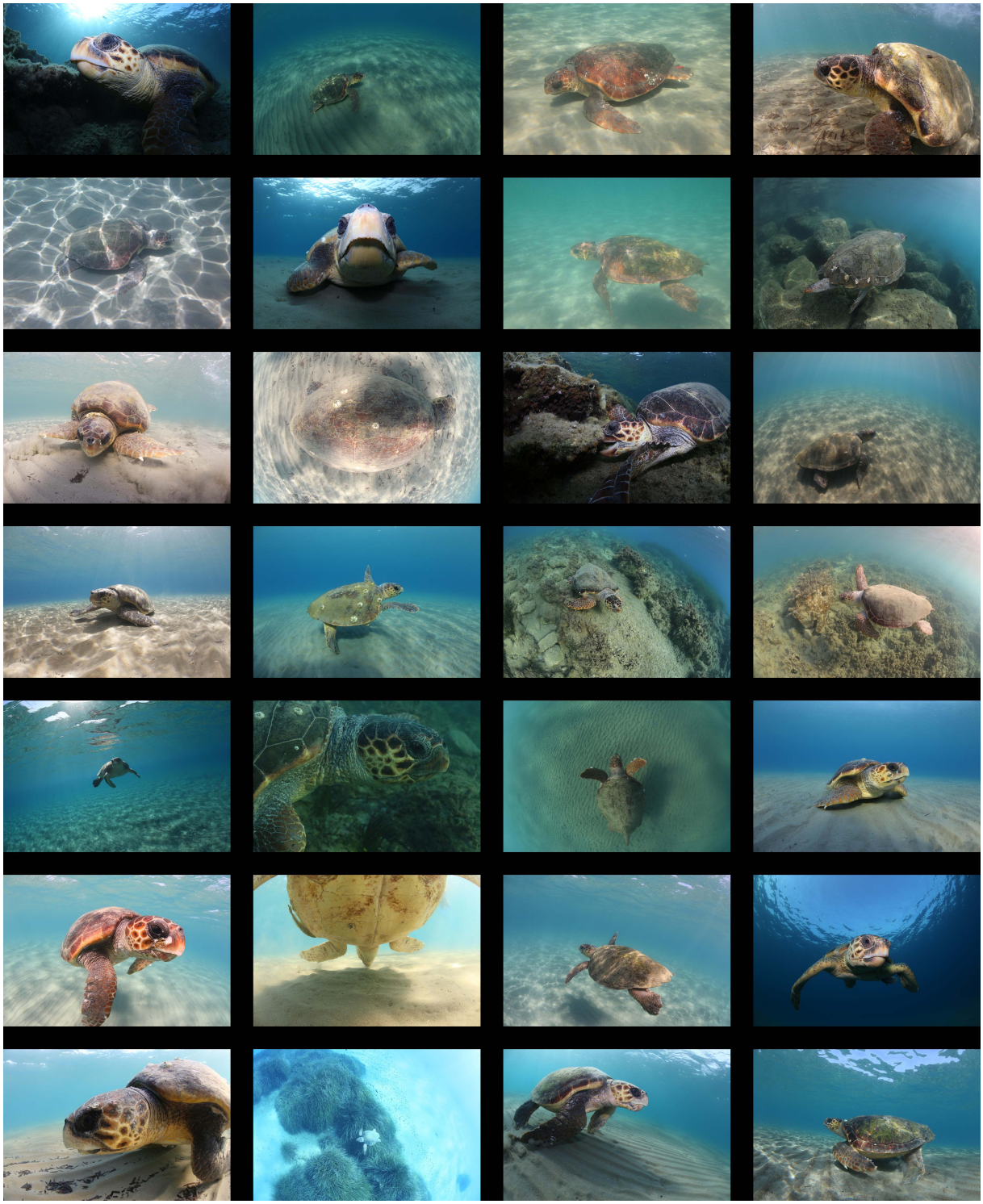


Figure 3.1: Examples of original photographs from the SeaTurtleID2022 dataset.

instead of learning an inner representation of each individual. Thus, a random split implicitly assumes that one will encounter the same factors in the future, which is highly unrealistic. On the other hand, timestamps allow for time-aware splits, where images from a time period are all in either the database or the query set. This leads to a more realistic case in which new factors are encountered in the future.

As shown in Chapter 2, just five publicly available datasets contain timestamps (see Table 3.1). From those, Cows2021 [61] and GiraffeZebraID [123] span only one month,

and WhaleSharkID [72] includes timestamps for only 9% of photographs. This leaves only two wildlife datasets with timestamps with span of at most two years. We introduce a novel dataset with photographs of loggerhead sea turtles (*Caretta caretta*) – the SeaTurtleID2022. The dataset was collected over 13 years and consists of 8729 high-resolution photographs of 438 unique individuals. Each photograph includes various annotations, e.g., identities, encounter timestamps, and body parts segmentation masks. To the best of our knowledge, the SeaTurtleID2022 is the longest-spanned public wild animal image dataset and the only public dataset of sea turtles with photographs captured in the wild. In contrast to existing datasets, the SeaTurtleID2022 allows for two realistic and ecologically motivated splits, instead of a *"time-unaware"* split:

- *time-proportional*: Splits images of each individual so that database images come from different encounters than the query images. This splitting method is detailed in Section 2.3.4.
- *time-cutoff*: Splits all images into database and query sets based on a specific time cutoff. This approach allows newly introduced individuals (i.e., those not previously recorded in the population) to appear in the query set, which is common in ecological studies. More details on this setting are provided in Section 2.3.4.

Dataset	**images**	**t-stamp**	**ind.**	**enc.**	**span**
Cows2021 [61]	8670	100%	179	3036	31
GiraffeZebraID [123]	6925	100%	2051	2494	12
MacaqueFaces [166]	6280	100%	34	494	525
BelugaID [13]	9304	100%	789	1557	785
WhaleSharkID [72]	7693	9%	98	424	1971
SeaTurtleID2022 (ours)	8729	100%	438	1221	4390

Table 3.1: Dataset statistics for all publicly available animal identification datasets with timestamps; number of photographs, percentage of photographs with timestamps, number of individuals and encounters, and dataset span in days.

Even though the SeaTurtleID2022 dataset is intended primarily as an animal identification benchmark, it can be used for the evaluation and testing of several fundamental problems, including: (i) object detection, (ii) instance segmentation, (iii) fully- and weakly supervised semantic segmentation, (iv) 3D reconstruction, and (v) concept drift analysis.

We stress that SeaTurtleID2022 lacks common drawbacks of other (human) identification datasets. In particular, face-id datasets typically contain low-resolution photographs, are restricted to limited poses, have limited time spans, and are either artificially generated [10], or collected by crawling the internet [73], raising privacy concerns.

2011: compact camera, no flash

2014: DSLR camera, no flash

2019: DSLR camera, with flash

Figure 3.2: Selected individual turtle (t023) from the SeaTurtleID2022 database, photographed with three different camera set-ups. Photographs taken with the DSLR camera are of higher quality, and the additional use of flash recovers the natural colouration of the animal. The photographs were cropped for illustration purposes.

3.2 The SeaTurtleID2022 dataset

This section describes the data collection process, annotation procedures, and key features of the SeaTurtleID2022 dataset.

3.2.1 Data collection

Location and species All photographs were taken in Laganas Bay, Zakynthos Island, Greece (37°43′N, 20°52′E), from 2010 until 2022; May–October. Laganas Bay is a main breeding site for the Mediterranean loggerhead sea turtles [102]. Female turtles (around 300 annually) are mainly migratory and visit the island to breed every 2–3 years [142]. On the other hand, certain individuals reside on the island, and they can be observed in consecutive years [121, 143]. Loggerheads are long-lived species, and they can have reproductive longevity of more than three decades [103], which can lead to long-span image recordings for specific individuals. Sea turtles are particularly amenable to photo-identification due to their scale patterns [141]. In particular, the polygonal scales in the lateral (side) and dorsal (top) sides of their heads are unique to every individual and remain stable throughout their lives [27].

This is illustrated in Figure 3.3, which shows examples of different visual appearances of the same individual sea turtles over long periods of time due to different factors like camera capture conditions and animal aging. The shapes of the facial scales remained stable, but other features have changed over time, like coloration, pigmentation, shape, and scratches.

Additionally, Figure 3.4 displays photographs of five individuals (one individual per row) showing the variability of the unique facial scale patterns of loggerhead sea turtles. The scales on the left and right sides of the head are different in a given individual, making it impossible to match them without any intermediate images.

Photographic procedure All photographs were captured underwater during snorkeling surveys from a distance ranging from 7 meters to a few centimeters using three cameras: (i) Canon IXUS 105 digital compact camera with a Canon underwater housing in 2010–2013, (ii) Canon 6D full-frame DSLR camera combined with a Sigma 15mm fisheye lenses and an Ikelite underwater housing in 2014–2017, and (iii) the same camera with an additional INON Z330 external flash in 2018–2022. The resolution ranges from

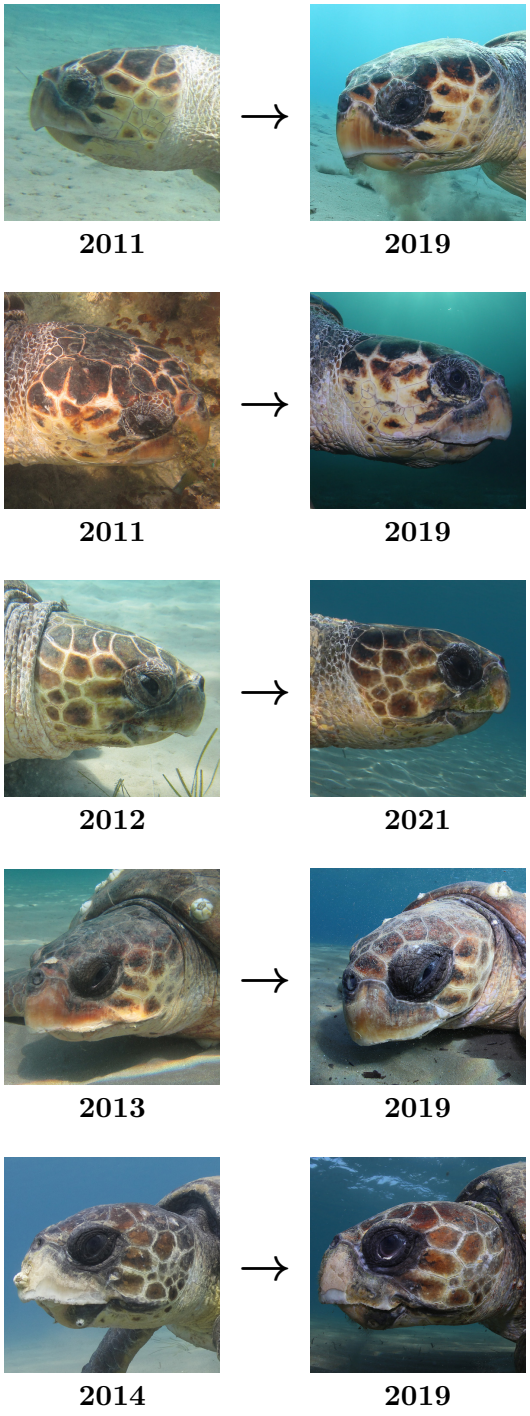


Figure 3.3: The long-term variation in the visual appearance of an individual sea turtle due to factors such as camera capture conditions and ageing. While the shapes of the facial scales remain unchanged, other features such as coloration, pigmentation, shape, and scratches change over time.

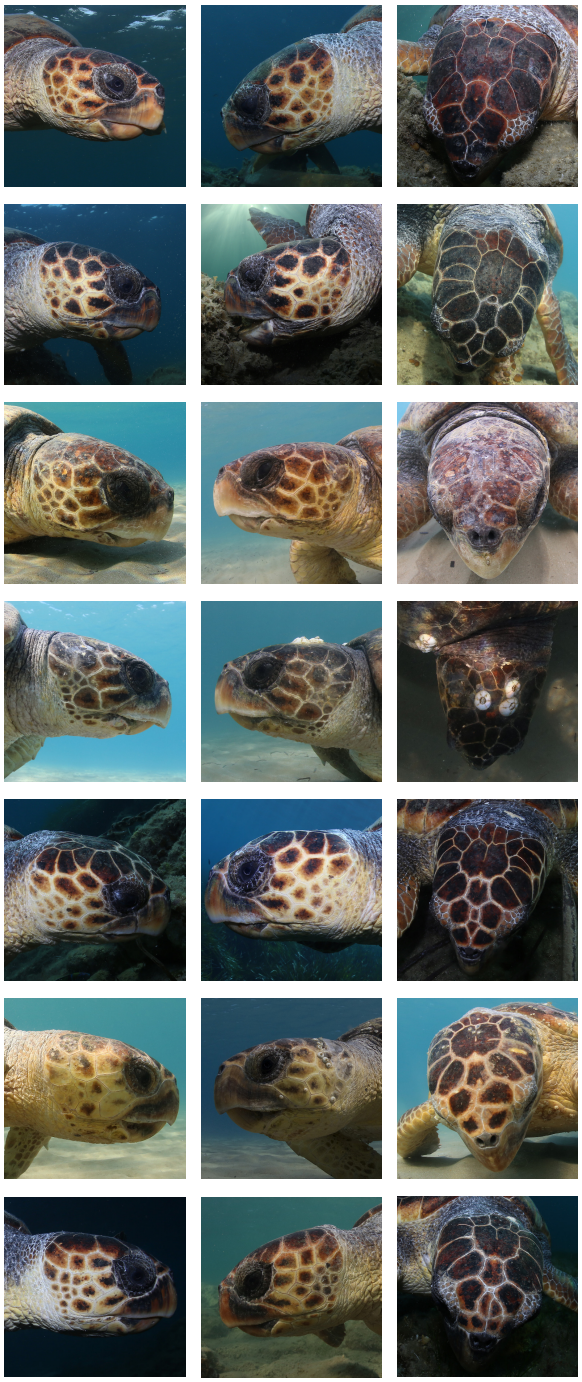


Figure 3.4: Examples of 7 individuals (one individual per row) that show the variability of unique facial scale patterns of loggerhead sea turtles. From left to right: right lateral facial scales, left lateral facial scales, dorsal head scales.

4000×3000 (Canon IXUS) to 5472×3648 pixels (Canon 6D) with an average of 5269×3564. The water depth ranged from 1 to 8 meters, with the vast majority of photographs taken less than 5 meters deep.

Photographs taken in 2014–2022 are generally of better quality due to the use of a more advanced camera and a shorter camera-subject distance. On the other hand, due to the use of fisheye lenses, barrel shape distortion can be noticeable, especially for close-up photographs. Finally, more natural colors were acquired using the external flash. In Figure 3.2, we display three images of the same individual – obtained by the three different camera set-ups – to highlight the resulting visual differences.

3.2.2 Dataset highlights

Large-scale in the wild dataset With 8729 photographs and 438 individuals, the dataset represents the most extensive publicly available dataset for sea turtle identification in the wild. The images are in original resolution and with various backgrounds. Approximately 90% of photographs have a size of 5472×3648 pixels, the average photograph size is 5269×3564 pixels, while the head occupies on average 635×554 pixels. Figure 3.5 shows the number of photographs for each individual. The majority of individuals ($\frac{272}{438}$) have at least ten photographs (depicted by the dashed line). Similarly, most individuals ($\frac{270}{438}$) were encountered at least twice. We note that this number is expected to increase in the following years since this dataset is updated annually.

Long time span & timestamps The dataset contains photographs continuously captured over 13 years from 2010 to 2022. In contrast to most existing animal datasets that are usually collected in controlled environments and/or over a short time span, the SeaTurtleID2022 dataset includes a timestamp (in dd:mm:yyyy format) for each photograph. Figure 3.6 (left) shows the number of encounters for each year, with a significantly larger number from 2015 onwards. We note that this is driven by an increasing data collection effort rather than reflecting actual annual recurrence. In Figure 3.6 (right), we show the number of newly observed individuals. Furthermore, Figure 3.6 (middle) shows the distribution of the 438 individuals with respect to the total number of observation years. A span of one year means that a turtle was photographed only in one year. Many turtles ($\frac{180}{438}$) were photographed in at least two different years, and 9 individual turtles spanned more than 9 years.

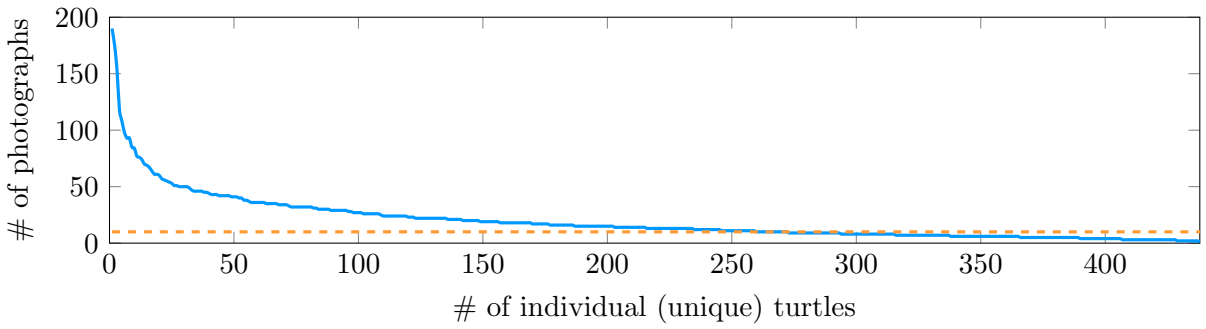


Figure 3.5: Number of photographs for each of the 438 turtles. The orange line corresponds to 10 photographs.

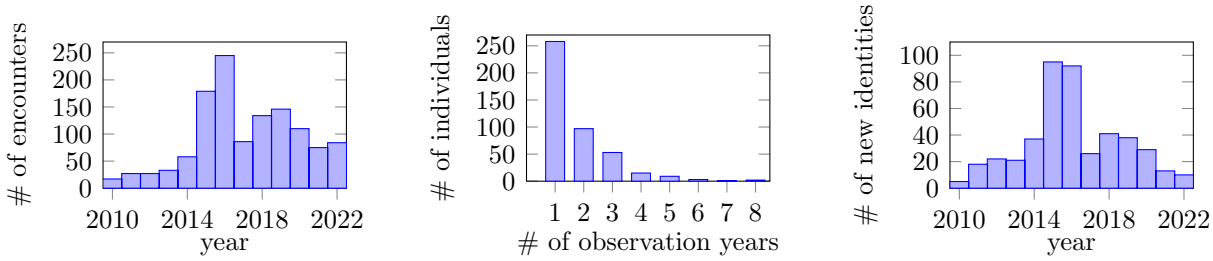


Figure 3.6: Time-related statistics within the SeaTurtleID2022 dataset: number of encounters per year (left), distribution of all individuals to the total number of observation years, i.e., recurrence of individuals (middle), and number of newly observed identities in each year (right).

Segmentation masks and bounding boxes Almost all photographs in the dataset have a visible head and/or flippers. Therefore we provide body parts annotations photographs as segmentation masks and bounding boxes. Apart from masks, we include orientation (left, right, top, top-right, top-left, front or bottom) for each head mask, and orientation (top or bottom) and location (front left/right or rear left/right) for flipper masks. Such annotations allow further development and evaluation of turtle identification methods or novel methods for object detection and semantic segmentation. All segmentation mask annotations were done semi-automatically using the Segment Anything [82] model integrated within the CVAT.

Multiple poses The dataset includes multiple images from different angles and, therefore, provides a ground for the challenging task of 3D animal reconstruction.

Comparison with ZindiTurtleRecall [155] For a better perspective, we compare the SeaTurtleID2022 with the ZindiTurtleRecall dataset, which is the only other publicly available sea turtle dataset. We stress that the latter dataset contains photographs in a controlled environment (a rehabilitation center) with no timestamps. We summarise all comparable aspects of both datasets in Table 3.2.

3.2.3 Dataset splits and subsets

Often, the identification datasets are split into a database (training) and a query set (test) randomly, which might result in unwanted data leakage and inflated performance. In other words, images from the same encounter might be in both sets. To illustrate the problem, we provide in Figure 3.8 four images of the same individual turtle, two captured in the same day in 2011 and two in the same day in 2021. While images from the same day are easy to match due to the same background and coloration, images from different days/years do not share it and therefore are significantly more challenging to match. To address this issue, we employ two realistic, ecologically motivated splits that use timestamps to prevent information leakage from the test set to the training set, as introduced in Section 2.3.4. The construction is further elaborated below. The dataset statistics, including the number of individuals and images, are listed in Table 5.3.

Time-proportion split We follow the definition of time-proportion split from Section 2.3.4. While constructing the split, we group all the data based on the date of acquisition

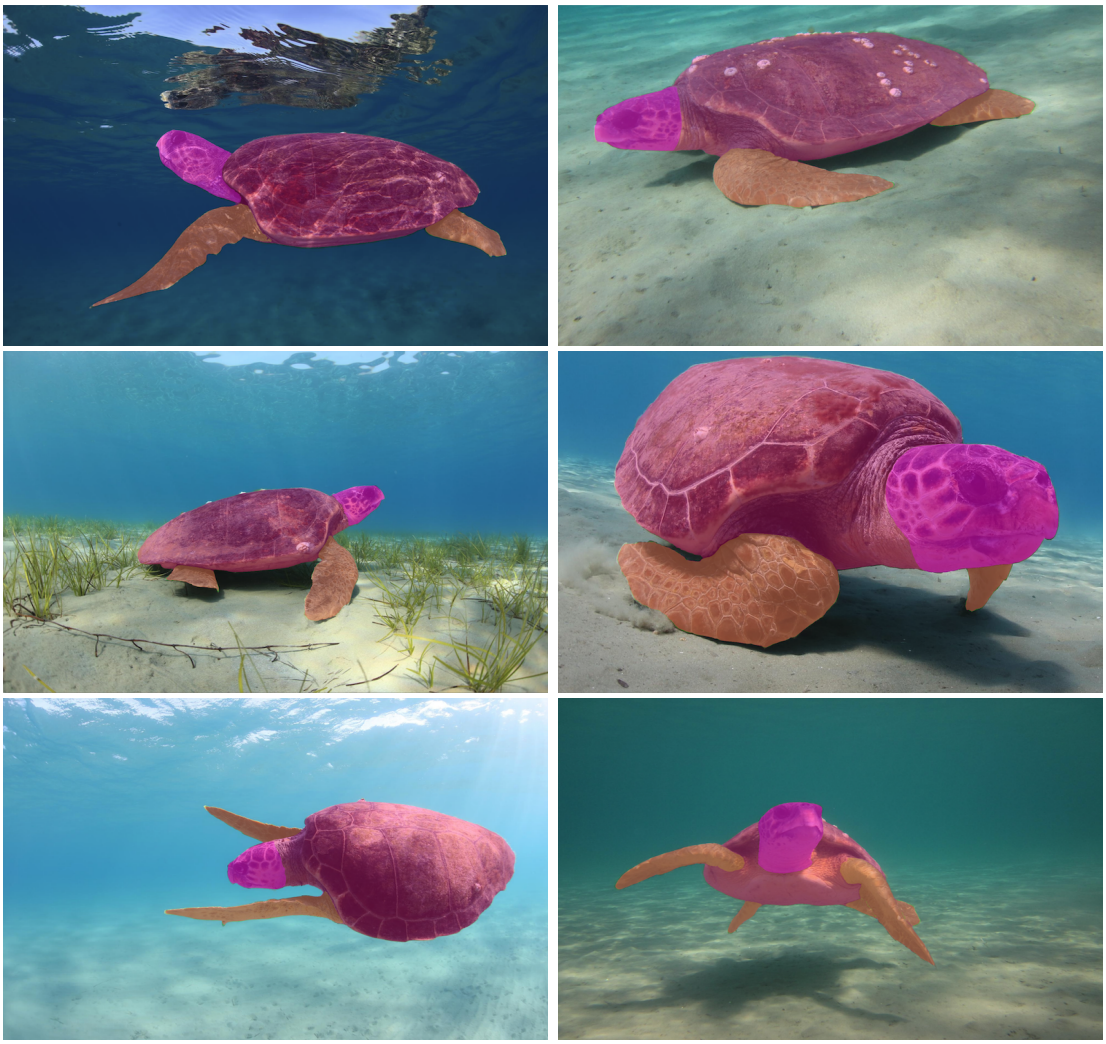


Figure 3.7: Examples of body parts (head, carapace, flippers) segmentation masks.

and split it in a time-aware fashion. Data from approximately 80% days are selected for the development set (training + validation), and the remaining days go to the test set. If an individual turtle was observed just once, it was kept for training. We provide 438 identities for training and 270 for testing. The development set was split into training/validation subsets using the same strategy.

Time-cutoff split We follow the definition of time-cutoff split from Section 2.3.4, where each subset (training/validation/test) contains all images within a time period based on cutoff years. During construction, we used the 2010–2018 period for training, the whole year of 2019 for the validation, and the 2020–2022 period for the test set. There are 357 identities in the training set and 151 in the test set. Out of the 151 identities, 51 are newly observed. A similar ratio (*new*/*known*) is naturally acquired in the validation set; 38 out of 83 are new identities.

Note: *The open-set split is much closer to the real-world animal identification settings than the closed-set problem. Therefore, the open-set split should be preferred for automated method evaluation over all datasets. In case closed-set evaluation is desired, then the time-aware split must be the preferred option over the random split.*

	SeaTurtleID2022	**ZindiTurtleRecall**
Sea turtle species	Loggerheads	Greens/Hawksbills
Images	8729	12803
Individuals	438	2265
Image average size	5269×3564	1382×1118
Head average size	635×554	1382×1118
Location	underwater	land (rehab. centre)
Allowed splits	*time-aware* & *open-set*	*random*
In the wild	✓	✗
Turtle segment	✓	✗
Head bbox	✓	✓
Head segment	✓	✗
Head orientation	✓	partially
Flipper segment	✓	✗
Flipper bbox	✓	✗
Timestamp	✓	✗

Table 3.2: Comparison with the ZindiTurtleRecall dataset.

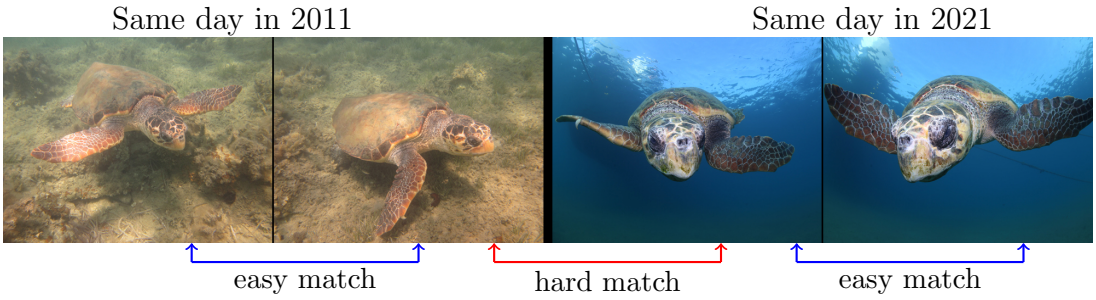


Figure 3.8: Unwanted background similarities in photographs from same/similar locations or time of observations.

	# of images		*# of identities*	
Subset	**closed-set**	**open-set**	**closed-set**	**open-set**
Training	4679	5303	438	357
Validation	1418	1118	91	83
Test	2632	2308	270	151

Table 3.3: Provided time-aware datasets split and their statistics.

Body-parts subsets Furthermore, we provide three subsets that cover various body parts, e.g., full-body, flippers, and heads, using crops from the original resolution. The number of data points differs for each body part, as some parts might not be visible. We used the time-aware closed-set and constructed part-based sets with the following number of training/test samples: (i) 6139 / 2650 full turtle bodies, (ii) 14849 / 6237 flippers, and (iii) 5956 / 2583 heads.

3.3 Sea turtle identification baselines

To establish a baseline performance on the SeaTurtleID2022 we perform various ablation studies using various methods defined in Chapter 2. In this section, we describe selected methods and all relevant hyperparameters.

3.3.1 Local feature-based methods

We study the performance of SIFT and more recent Superpoint[49] descriptors on the proposed dataset. We have developed a straightforward algorithm (inspired by Dunbar et al. [53]) based on local descriptors matching[1]. First, we extract a set of keypoints and their corresponding descriptors for each image. Second, for all possible training-test image pairs, we calculate the distance between their descriptors. Third, all potentially false matches are filtered out using a ratio test and threshold; the optimal values (0.2 for SIFT, 0.6 for Superpoint) for the ratio test thresholds were obtained from the training set. At last, we predict an identity using the training label with maximal similarity score, calculated as an absolute number of correspondences. We opt not to use alternative approaches, such as RANSAC or SuperGlue, as they add significant computational overhead and provide just a small improvement [53, 125].

3.3.2 Metric learning

We compare local feature matching with the deep metric learning methods ArcFace [47] and Triplet Loss [144]. For more details on these methods, we refer the reader to Section 2.4. For baseline performance evaluation of the metric learning methods, we use a Swin-B,[98] backbone with default training hyperparameters. Models are optimized for 100 epochs using a learning rate of 0.01, a cosine annealing schedule, and a mini-batch size of 128. All images are pre-processed using the Random Augment method

For inference, we adopt an image retrieval-based approach to animal identification, utilizing a k-NN classifier in a deep embedding space, as described in Section 2.3.2. For each test image x, we retrieve its k most similar identities from the training set and assign the class based on the most frequent identity among them.

3.3.3 Random vs. time-aware splits

To showcase the unintended performance overestimation when a random dataset split is used, we compare the performance of newly proposed time-aware splits (open and closed) with their random counterparts. The random split is obtained by randomly shuffling the time-aware split for each identity separately. This ensures a fair comparison between the split with the same training/validation/test ratio. We used the entire image and different body parts in this experiment. We use an ArcFace loss with the Swin-B backbone and input size of 224×224.

[1]For SIFT we use default parameters and OpenCV implementation; for Superpoint, we use default parameters and this implementation.

3.4 Baseline Results

In this section, we provide (i) baseline results for body-part segmentation and identification achieved over the newly proposed dataset, (ii) qualitative and quantitative evaluation to show the importance of the time-aware splits, and (iii) performed ablation studies to select the most viable approach for sea turtle identification. Based on extensive experiments (Section 3.5.1) with different k values for k-NN matching, we predict an identity using k-NN, with $k = 1$.

Local vs deep features Comparing local descriptors with metric learning approaches showed superior performance of metric learning on our dataset and seven other datasets with patterned species. In most cases, the metric learning approaches outperformed the Superpoints by more than 20%. Furthermore, if we compare local descriptor methods, the Superpoints method is a better fit for animal identification. A detailed comparison is listed in Table 3.4.

Dataset	SIFT	Superpoint	ArcFace	Triplet
BelugaID [13]	1.1	2.4	18.2	20.5
HumpbackWhaleID [77]	11.7	11.8	52.5	43.9
NDD20 [152]	17.1	30.0	59.1	29.9
NOAARightWhale [133]	6.5	15.3	23.5	5.4
WhaleSharkID [72]	4.3	22.9	28.6	32.5
ZindiTurtleRecall [155]	17.9	25.7	45.8	19.1
SeaTurtleID2022 (ours)	8.4	20.2	34.7	25.7

Table 3.4: Local and deep feature methods performance comparison (accuracy) for full images. Time-aware closed-set split. Input size 224×224. For metric learning, a Swin-B backbone was used.

Body parts performance In addition to setting overall full-body turtle performance, we explored the importance of various body parts, revealing their relative significance. In contrast to the findings of [108], our results highlight the key role of the turtle's *head* in sea turtle identification. Focusing solely on the *head* increased the absolute performance by 34.5% compared to the full body. Furthermore, we show that the *flippers* appear as the less influential body part for in-the-wild identification using metric learning[2]. The full comparison is provided in Table 3.5.

Encounter based prediction Available timestamps enable grouping image-based predictions into encounters, where all images taken within a similar timeframe (e.g., within a day) are considered part of the same encounter. This approach allows for a combined prediction using all images of an individual from a single encounter rather than identifying each image separately. By applying simple majority voting to merge image-based predictions, we significantly improved performance across all body parts, with head identification accuracy increasing by 19.2% (see Table 3.5).

[2]For the flippers performance evaluation, we choose the closest (based on cosine similarity) identity using all available flippers on a given image.

3.4.1 Random vs time-aware splits

The performance comparison of two ArcFace-trained feature extractors on the random and time-aware splits of the SeaTurtleID2022 dataset validated our hypothesis about unwanted performance inflation related to training-to-test data leakage. Results listed in Table 3.5 demonstrate that the random split results (in terms of accuracy) were higher by 42.2%, 53.8%, 45.8%, and 18% for full image, and flippers, body, and head crops, respectively.

	Split	Full image	Flippers	Turtle	Head
Images	Time-aware	17.1	12.2	34.7	69.2
Encounters	Time-aware	–	21.4	48.6	88.4
Images	Random	*59.4*	*66.0*	*80.5*	*87.2*

Table 3.5: Random split accuracy inflation on SeaTurtleID2022 (closed-set). Encounter- vs image-based; Swin-B + ArcFace.

Performance inflation analysis To further elaborate on the performance inflation, we conducted an additional identification experiment using (i) images with redacted backgrounds, showing only the turtle in the foreground, and (ii) images with redacted foregrounds, displaying only the background. With the redacted background, the model's performance remains relatively comparable to the full image performance in both scenarios. Contrarily, in the case of redacted foreground, the model trained on a random split exhibits comparable performance to that achieved on the full images. However, the performance for the model trained on a time-aware dropped significantly in performance relative to the full images, achieving only 3.9% accuracy. See results in Table 3.6.

Split	Full image	Background	Foreground
Random	59.4	45.1	59.5
Time-aware	17.1	3.9	14.3
Δ	*+42.2*	*+41.2*	*+45.2*

Table 3.6: Random split accuracy inflation on the SeaTurtleID2022 (closed-set). Swin-B + ArcFace; 224 × 224.

Furthermore, we qualitatively demonstrate overfitting to the background using Grad-CAM++ [32] and visualizing identity activations based on the cosine similarity between the embeddings of the two images. We selected two similar images with noticeable backgrounds from the same encounter that are in the test set for both random and time-aware splits. In Figure 3.9, we illustrate that the model trained on the random split learns to utilize background features, whereas the model trained using the time-aware approach concentrates on the turtle's features.

3.4.2 Body-parts segmentation baselines

The SeaTurtleID2022 dataset comes along with instance segmentation annotations; thus, it might be used as a benchmark for instance segmentation or object detection. To

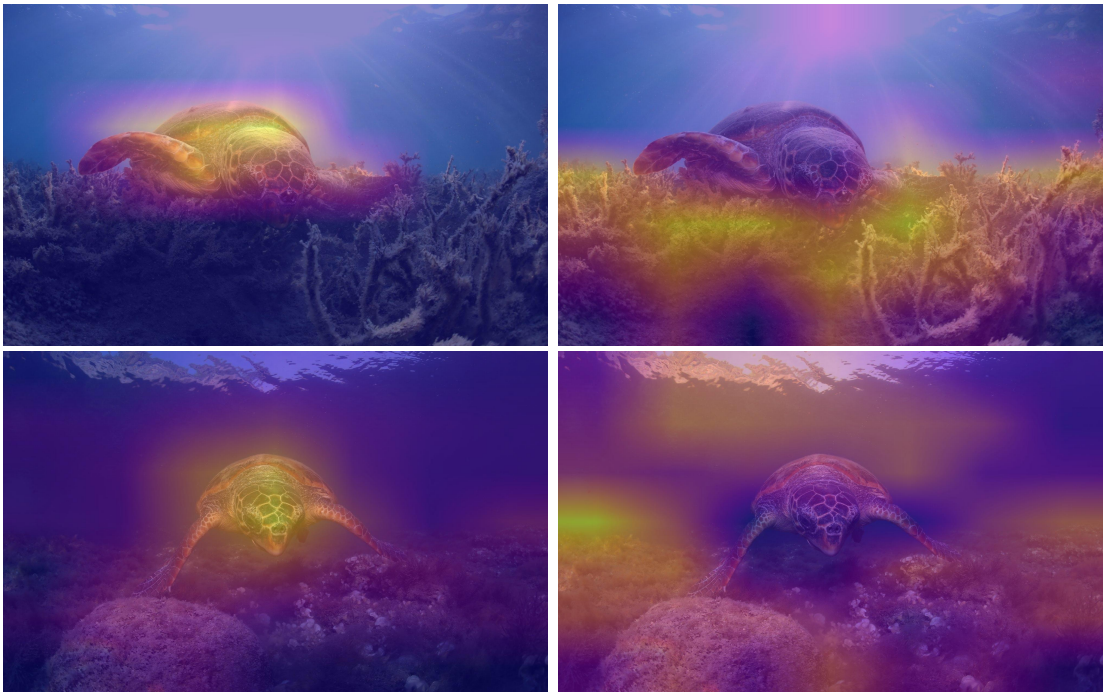


Figure 3.9: Qualitative evaluation demonstrating overfitting to the background on random split using Grad-CAM++. Identity-based activations for (left) time-aware and (right) random split.

set the baseline performance for the turtle body parts (head, flipper, and full-body) segmentation, we evaluate three distinct architectures, including the standard Mask R-CNN [68], the Hybrid Task Cascade (HTC) [35], and the state-of-the-art transformer-based Mask2Former [39]. We combine the three detection methods with two backbones, ResNet-50 [67] and Swin-B transformer [98] using the MMDetection [36] framework. While training, both backbones were initialized from publicly available ImageNet-1k weights using the default implementation and hyperparameters setting. All models were fine-tuned for 12 epochs with a step-wise learning rate schedule. Experiments are conducted on both time-aware splits.

Generally, all selected methods evaluated on the SeaTurtleID2022 achieved a competitive performance (in terms of coco mAP) suitable for the following task, i.e., turtle identification. While the best-performing model – *Mask2Former with Swin-B backbone* – achieved a coco mAP of 0.896, the worst-performing model – *Mask R-CNN with ResNet-50 backbone* – achieved an mAP of 0.865. Even though the Mask2Former approach showed better overall performance, the HTC method performed better on heads that are important for accurate identification. The full performance comparison is available in Table 3.7

3.5 Ablations studies

3.5.1 *k*-NN classifier ablation

We conducted additional experiments to find an optimal value of k for the animal identification using the k-NN classifier. Besides SeaTurtleID2022 (head and full-body versions), we evaluated the experiments on BelugaID, NDD20, WhaleSharkID, Humpback-

WhaleID, NOAARightWhale and ZindiTurtleRecall datasets. We used embeddings from the ArcFace-trained model.

Our findings indicate that opting for a smaller k value yields better results, with k=1 being a reasonable choice in all cases. We attribute this phenomenon to the significant class imbalance present in animal datasets. As k increases, identities with higher prior probability overwhelm the classification results, i.e., for larger k values, there are often just a few samples for the less frequent identities. On the SeaTurtleID2022 dataset (head and full-body) the performance in terms of accuracy significantly decreased from 69.2% at $k = 1$ to 55.0 % at $k = 100$. A similar, though less severe, drop in performance was also noticeable in other datasets. We depict the relationship between accuracy and values of k in Figure 3.10.

3.5.2 Cross-entropy loss ablation

To further elaborate on the performance inflation related to random split, we tested various deep learning backbone architectures optimized using softmax cross-entropy. Table 3.8 shows the performance of five architectures on two splits of the SeaTurtleID2022 dataset: time-aware and random. The results show that performance inflation persists across all tested architectures, even when using cross-entropy loss.

3.5.3 Time-Aware Splitting Across Datasets

We further test and demonstrate the need for time-aware splits on additional datasets that include timestamps. Table 3.9 presents results from experiments on three other datasets that support time-aware splitting, demonstrating that performance inflation is not unique to the SeaTurtleID2022 dataset but occurs across different datasets. In all cases, the results from the random split are undesirably inflated, outperforming those from the time-aware split. We used the Swin-B/p4w7 model, trained on a 50/50 training-test split, and all images were resized to match the modelś input size of 224x224.

	Method	mAP	*head*	*turtle*	*flippers*
ResNet-50	Mask R-CNN	0.865	0.838	0.910	0.848
	HTC	0.868	0.842	0.912	0.852
	Mask2Former	0.892	0.822	0.977	0.876
Swin-B	Mask R-CNN	0.871	0.845	0.919	0.85
	HTC	0.880	0.860	0.923	0.857
	Mask2Former	0.896	0.829	0.975	0.883

Table 3.7: Instance segmentation performance of selected *backbone* and *head* architectures over the SeaTurtleID2022. Closed-set split.

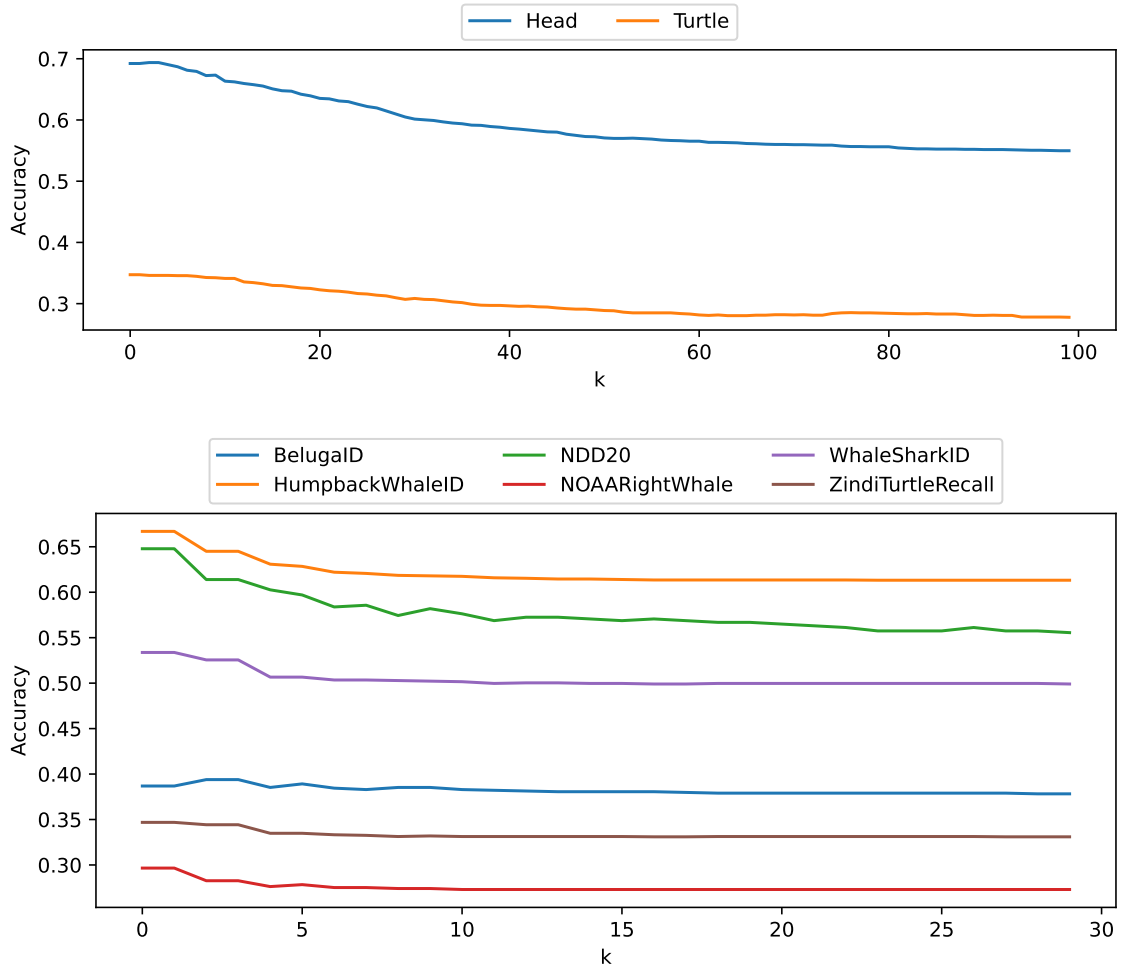


Figure 3.10: **Effect of k on performance**. We display the classification accuracy of k-NN classifier with ArcFace embeddings for various k values. Different body parts (e.g. head and full-body) performance on the SeaTurtleID2022 dataset (top) and selected animal identification datasets (bottom).

Backbone	**Time-aware closed-set**	*Random split*
ResNeXt-50	38.6%	*63.4%*
EfficientNet-B0	39.9%	*76.5%*
ConvNeXt-B	47.2%	*78.5%*
ViT-Base/p32	45.2%	*82.5%*
Swin-B/p4w7	47.6%	*83.2%*

Table 3.8: Performance inflation (accuracy) with different backbones fine-tuned with softmax cross-entropy.

Dataset	**Time-aware closed-set**	*Random split*
BelugaID	7.8%	*12.1%*
GiraffeZebraID	2.1%	*30.1%*
MacaqueFaces	91.1%	*98.9%*

Table 3.9: Performance inflation (accuracy) with different datasets.

Chapter 4

Animal identification tookit

Standardization of algorithmic procedures, evaluation metrics, and dataset utilization is critical for advancing animal identification research. However, inconsistencies in these aspects across the literature make it challenging to compare results and reproduce findings, slowing progress in the field. It is, therefore, essential to categorize and re-evaluate general identification approaches, connect them to real-world scenarios, and provide recommendations for appropriate algorithmic setups in specific contexts. By quantitatively assessing the approaches employed in various studies, we aim to identify trends and provide insights into the most effective techniques for different scenarios.

Similarly, as in other fields, the development of methods and datasets for automated animal identification has been influenced by the progress in machine learning. Currently, many studies exist, although the differences in terms of their approach, prediction output, and evaluation methodologies result in several drawbacks.

- **Firstly**, methods are usually inspired by trends in machine learning rather than being motivated by real-world identification scenarios. A prominent example is performing classification tasks on a closed-set, which is typical for benchmarking in deep learning but is, in general, not realistic in ecology, as new individuals are constantly being recruited to populations.

- **Second**, many studies focus on a single dataset and develop species-specific methods evaluated on the given dataset rather than on a family of datasets [12, 163, 64, 94, 52, 5], making reproducibility, transferability, and generalization challenging.

- **Third**, datasets are poorly curated and usually include unwanted training-to-test data leakage, which leads to inflated performance expectations.

All this leads to the repetition of poor practices both in dataset curation and method design. As such, much of the current research suffers from a lack of unification, which, we argue, constitutes an obstacle to further development, evaluation, and applications to real-world situations.

To address these issues, we have developed an open-source toolkit intended primarily for ecologists and computer-vision / machine-learning researchers. In this chapter, besides the description of the main features of our tool, we list all publicly available animal identification datasets.

4.1 The WildlifeDatasets toolkit

One of the current challenges for the advancement of animal identification methods is the fact that datasets are scattered across the literature and that adopted settings and developed algorithms heavily focus on the species of interest. In order to facilitate the development and testing of identification methods across multiple species in scale and evaluate them in a standardized way, we have developed the Wildlife Datasets toolkit consisting of two Python libraries – WildlifeDatasets and WildlifeTools[1]. Both libraries are documented in a user-friendly way; therefore, it is accessible to both animal ecologists and computer vision experts. Users just have to provide the data and select the algorithm. Everything else can be done using the toolkit: extracting and loading data, dataset splitting, identity matching, evaluation, and performance comparisons. Experiments can be done over one or multiple datasets fitting into any used specified category, e.g., size, domain, species, and capturing conditions. Below, we briefly describe the core features and use cases of both libraries.

All publicly available animal datasets at hand

The first core feature of the WildlifeDatasets toolkit allows downloading, extracting, and pre-processing all 42 publicly available animal datasets[2] (refer to Table 4.1) in a unified format using just a few lines of Python code. Additionally, users can quickly overview and compare images of the different datasets and their associated metadata, e.g., image samples, number of identities, timestamp information, presence of segmentation masks/bounding boxes, and general statistics about the datasets. This feature decreases the time necessary for data gathering and pre-processing tremendously. Recognizing the continuous development of the field, we also provide user-friendly options for adding new datasets.

Implementation of advanced dataset spliting

Apart from the datasets at hand, the toolkit has built-in implementations for all dataset training/validation/test splits corresponding to the different settings, including (i) *closed-set* with the same identities in training and testing sets, (ii) *open-set* with a fraction of newly introduced identities in testing, and (iii) *disjoint-set* with different identities in training and testing. In cases where a dataset contains timestamps, we provide so-called time-aware splits where images from the same period are all in either the training or the test set. This results in a more ecologically realistic split where new factors, e.g., individuals and locations, are encountered in the future [122].

Accessible feature extraction and matching

Apart from the datasets, the WildlifeDatasets toolkit provides the ability to access multiple feature extraction and matching algorithms easily and to perform identification on the spot. We provide a variety of local descriptors, pre-trained CNN- and transformer-based descriptors, and different flavors of the newly proposed foundation model – MegaDescriptor. Below, we provide a short description of all available methods and models.

[1] Both libraries are available online on GitHub.
[2] Based on our research at the end of October 2024.

Local descriptors Due to extensive utilization among ecologists and state-of-the-art performance in animal identification, we have included selected local feature-based descriptors as a baseline solution available for deployment and a direct comparison with other approaches. Within the toolkit, we have integrated our implementations of standard SIFT and multiple deep learning-based descriptors such as SuperPoint [49], DISK [157], and ALIKED [174]. Besides, we have integrated local feature matching algorithms based on deep learning, such as LightGlue[96] and LoFTR[147].

Pre-trained deep-descriptors Besides local descriptors, the toolkit allows to load any pre-trained model available on the HuggingFace hub and to perform feature extraction over any identification datasets. We have accomplished this by integrating the Timm library [165], which includes state-of-the-art CNN- and transformer-based architectures, e.g., ConvNeXt [99], ResNext [167], ViT [51], and Swin [98]. This integration enables both the feature extraction and the fine-tuning of models on the animal identification datasets.

MegaDescriptor Furthermore, we provide the first-ever foundation model for individual identification within a wide range of species – MegaDescriptor (described in Chapter 5) – that provides state-of-the-art performance on all datasets and outperforms other pre-trained models such as CLIP and DINOv2 by a significant margin. In order to provide the models to the general public and to allow easy integration with any existing wildlife monitoring applications, we provide multiple MegaDescriptor flavors, e.g., Small, Medium, and Large.

Matching Next, we provide a user-friendly high-level API for matching query and reference sets, i.e., to compute pairwise similarity. Once the matching API is initialized with the identity database, one can simply feed it with images, and the matching API will return the most visually similar identity and appropriate image.

Community-driven extension

Our toolkit is designed to be easily extendable, both in terms of functionality and datasets, and we welcome contributions from the community. In particular, we encourage researchers to contribute their datasets and methods to be included in the WildlifeDataset. The datasets could be used for the development of new methods and will become part of future versions of the MegaDescriptor. This collaborative approach aims to further drive progress in the application of machine learning in ecology. Once introduced in communities such as LILA BC or AI for Conversation Slack[3], the toolkit has a great potential to revolutionize the field.

Online Documentation – Dataset samples and tutorials

We provide extensive documentation to give users a better orientation within the WildlifeDatasets toolkit and available features. It covers a wide range of use cases of the toolkit, including a guide to installation and dataset downloading, tutorials, and how to contribute. Notably, the documentation includes a detailed description of the datasets, including image samples.

[3]With around 2000 members; experts on ecology and machine learning.

4.2 Available datasets

In this section, we provide a brief description of the datasets included in the library. The description focuses on the dataset sources and how it was collected. General information, such as the number of images and identities, is summarized in Table 4.1.

AAUZebraFishID [22] Three zebrafish were placed in a small clear glass tank, and a video was captured. The authors used a careful setup to ensure that the fish are approximately the same distance from the camera and that the lighting conditions are good. Frames were extracted, and the three fish were manually tracked using bounding boxes. This was repeated two times with two different sets of fish, resulting in images of six fish in total.

AmvrakikosTurtles [1] This data set consists of photographs of Mediterranean loggerhead sea turtles taken at the Amvrakikos Gulf, Greece, which is a well-known foraging site for adult and juvenile turtles. Photographs were collected as part of a long-term capture-mark-recapture project conducted by ARCHELON, the Sea Turtle Protection Society of Greece. The turtles were captured from a boat using the sea turtle rodeo technique, and, among other data collected, photographs of the head sides were taken while the animal was on the boat. In all photographs, either the whole side of the head was fully shaded or fully illuminated by the sun. All photographs in this dataset were taken during the summer months between 2014 and 2022, using a selection of different digital cameras of varying optical resolution.

FriesianCattle2015 [8] + **FriesianCattle2017** [7] + **AerialCattle2017** [7] + **OpenCows2020** [9] + **Cows2021** [61] These datasets were created by one group. They capture Holstein-Friesian cows from an aerial standpoint. All images were extracted from videos. FriesianCattle2015 and FriesianCattle2017 were obtained by filming cows exiting a milking file. AerialCattle2017 was captured by a drone in an outdoor agricultural field environment. OpenCows2020 combines these datasets. Since the distance of the camera ranges from approximately 4m (FriesianCattle2015) to 25m (AerialCattle2017), it is relatively easy to separate these datasets. Moreover, no individual cow seems to be present in both image acquisitions. Cows2021 depicts the cows in a similar way as FriesianCattle2015 when the camera pointed downwards from 4m above the ground over a walkway between the milking parlor and holding pens. Some of the datasets are provided with videos, and besides cow identification, they also aim at cow detection and localization.

ATRW [94] The ATRW (Amur Tiger Re-identification in the Wild) dataset was collected with the help of WWF in ten zoos in China. The images were extracted from videos. Besides tiger identification, the dataset can also be used for tiger detection and pose estimation.

BelugaID [13] BelugaID is a high-quality dataset published by WildMe. It contains labeled images of beluga whales collected as part of a collaborative effort focused on data collection and population modeling in the Cook Inlet off the coast of Alaska from 2016 to 2019. The dataset includes pre-cropped, high-quality images taken primarily from a top-down perspective.

Name	Year	# Images	# Identities	Timestamp	In-the-wild	Muspecies
AAUZebraFishID [22]	2020	6672	6	✗	✗	✗
AerialCattle2017 [7]	2017	46340	23	✗	✗	✗
AmvrakikosTurtles [1]	2024	200	50	✗	✓	✗
ATRW [94]	2019	5415	182	✗	✗	✗
BelugaID [13]	2022	5902	788	✓	✓	✗
BirdIndividualID [56]	2019	51934	50	✗	✗	✓
CatIndividualImages [30]	2020	13021	509	✗	✗	✗
CTai [57]	2016	4662	71	✗	✓	✗
CZoo [57]	2016	2109	24	✗	✗	✗
Chicks4FreeID [81]	2024	1146	50	✗	✗	✗
CowDataset [93]	2021	1485	13	✗	✗	✗
Cows2021 [61]	2021	8670	181	✓	✗	✗
DogFace [113]	2019	8363	1393	✗	✗	✗
Drosophila [137]	2018	∼2.6M	60	✗	✗	✗
ELPephants [85]	2019	2078	274	✓	✓	✗
FriesianCattle2015 [8]	2016	377	40	✗	✗	✗
FriesianCattle2017 [7]	2017	940	89	✗	✗	✗
GiraffeZebraID [123]	2017	6925	2056	✓	✓	✓
Giraffes [107]	2021	1393	178	✗	✓	✗
HappyWhale [34]	2022	51033	15587	✗	✓	✗
HumpbackWhaleID [77]	2019	15697	5004	✗	✓	✗
HyenaID2022 [153]	2022	3129	256	✗	✓	✗
IPanda50 [161]	2021	6874	50	✗	✗	✗
LeopardID2022 [153]	2022	6806	430	✗	✓	✗
LionData [50]	2020	750	94	✗	✓	✗
MacaqueFaces [166]	2018	6280	34	✓	✗	✗
MPDD[70]	2023	1657	191	✗	✗	✗
NDD20 [152]	2020	2657	82	✗	✗	✗
NOAARightWhale [133]	2015	4544	447	✗	✓	✗
NyalaData [50]	2020	1942	237	✗	✓	✗
OpenCows2020 [9]	2020	4736	46	✗	✗	✗
PolarBearVidID[176]	2023	138363	13	✗	✗	✗
SealID [116]	2022	2080	57	✗	✓	✗
SeaStarReID2023[160]	2023	2187	95	✗	✗	✓
SeaTurtleID [122]	2022	7774	400	✓	✓	✗
SeaTurtleID2022 [3]	2024	8729	438	✓	✓	✗
SMALST [178]	2019	12850	10	✗	✗	✗
SouthernProvinceTurtles[1]	2024	481	51	✗	✓	✗
StripeSpotter [89]	2011	820	45	✗	✓	✗
WhaleSharkID [72]	2020	7693	543	✓	✓	✗
ZindiTurtleRecall [155]	2022	12803	2265	✗	✓	✗

Table 4.1: **Publicly available animal re-identification datasets**. We list all datasets for animal re-identification and their relevant statistics, e.g., number of images, identities, etc. All listed datasets are available for download in the WildlifeDatasets toolkit.

BirdIndividualID [56] BirdIndividualID is a collection of three separate bird datasets: sociable weavers at Benfontein Nature Reserve in Kimberley, South Africa, wild great tits in Möggingen, Germany, and captive zebra finches at the same place. The individuals of sociable weavers and great tits were fitted with PIT tags as nestlings or when trapped in mist nets as adults. The collection of labeled pictures in the wild was automated by combining RFID technology, single-board computers (Raspberry Pi), Pi cameras, and artificial feeders. The authors fitted RFID antenna to small perches placed in front of bird feeders filled with seeds. The RFID data logger was then directly connected to a Raspberry Pi with a camera. When the RFID data logger detected a bird, it sent the individual's PIT-tag code to the Raspberry Pi, which took a picture. The cages of captive zebra finches were divided into equally sized partitions with a net, allowing us to take pictures of individual birds without completely socially isolating them. Besides the full images, they provided segmentated images of all birds. This is the only dataset where authors admitted that some of the labels are wrong. This stemmed from the automatic procedure of labeling, where multiple birds sometimes entered the artificial feeder, and the camera took a picture of the wrong bird. They manually checked the sociable weaver images, and 4.4% images were confirmed to be mislabelled.

CatIndividualImages [30] The images were acquired using regular digital cameras or smartphones. The resolutions of the images ranged between 195 x 261 and 4608 x 3453 pixels. Most images of cats were acquired at public animal shelters (Taipei City Animal Protection Office, Taipei, Taiwan; and New Taipei City Government Animal Protection and Health Inspection Office, New Taipei, Taiwan) and private animal shelters in Taipei. The remaining images were collected from social media.

Chicks4FreeID [81] The Chicks4FreeID dataset contains top-down view images of individually segmented and annotated chickens (with roosters and ducks also possibly present and labeled as such). 11 different coops with 54 individuals were visited for manual data collection. Each of the 677 images depicts at least one chicken. The identities of the 50 chickens, 2 roosters and 2 ducks were annotated for a total of 1270 animal instances. Annotation additionally contains visibility ratings of "best", "good", and "bad" for each animal instance.

CTai [100] + **CZoo** [100] CTai and CZoo datasets are extended subsets of the datasets [100]. Both contain cropped chimpanzee faces. CZoo originates from a collaboration between the authors and animal researchers in Leipzig. The provided images are of high quality, well exposed, and can be taken without strong blurring artifacts. The images are complemented by biologically meaningful keypoints (centers of eyes, mouth, and earlobes) together with information about age and gender. CTai consists of recordings of chimpanzees living in the Taï National Park in Côte d'Ivoire. The image quality differs heavily, and the annotation quality of additional information is not as high as that of CZoo. CTai contains typos in six individuals (such as Woodstiock instead of the correct Woodstock), which we corrected. The unknown individuals were labeled as "Adult", which we fixed as well.

CowDataset [93] The dataset consists of 3772 lateral-view images of 13 Holstein cows captured in June 2021 at Dongfeng Cattle Farm, Liaoyuan City, Jilin Province. The images were taken using a Canon EOS 5D Mark II camera with a maximum resolution

of 5616 × 3744 pixels. The dataset reflects real farming conditions, as the pictures were taken against complex backgrounds.

DogFace [113] The dataset includes photographs taken by the researchers as well as images sourced from non-profit pet adoption websites such as Streunerhilfe, Tiko, Pfotenhilfe, La SPA, Tieronline, and Animal-happyend. Only dogs with more than five pictures were included, resulting in a final collection of 3148 images representing 485 individual dogs.

Drosophila [137] Twenty drosophila flies were collected a few hours after eclosion and housed separately. On the third day, post-eclosion flies were individually mouth pipetted into a circular acrylic arena, illuminated with overhead LED bulbs, and filmed in grayscale. This was repeated for three consecutive days. Since the sampling frequency from videos was high, this generated several million images. However, the differences between these images are small.

ELPephants [85] This elephant dataset was provided by researchers from the Elephant Listening Project (ELP) at Cornell University Ithaca, who are conducting research on forest elephants visiting the Dzanga bai clearing in the Dzanga-Ndoki National Park in the Central African Republic. It was devised for the identification of elephants that have been documented before. The images have been taken over a range of about 15 years.

GiraffeZebraID [123] GiraffeZebraID contains images of plains zebra and Masai giraffe taken from a two-day census of Nairobi National Park with the participation of 27 different teams of citizen scientists and 55 total photographers. The photographers were recruited both from civic groups and by asking for volunteers at the entrance gate in Nairobi National Park. All volunteers were briefly trained in a collection protocol and tasked to take pictures of animals within specific regions and from specific viewpoints. These regions helped to enforce better coverage and prevent a particular area from becoming oversampled. Only images containing either zebras or giraffes were included in this dataset. All images are labeled with viewpoints and possibly rotated bounding boxes around the individual animals. All of the images in the dataset have been resized to have a maximum dimension of 3,000 pixels.

HappyWhale [34] + **HumpbackWhale** [77] + **NOAARightWhale** [133] HappyWhale, HumpbackWhale, and NOAARightWhale are datasets of various whale species. They are a product of the multi-year collaboration of multiple research institutions and citizen scientists. All these datasets were released as Kaggle competitions to make it easy and rewarding for the public to participate in science by building innovative tools to engage anyone interested in marine mammals. The whales were photographed during aerial surveys. HumpbackWhale is the most uniform dataset with a clear view of the whale tail above the water. NOAARightWhale contains images of submerging whales. HappyWhale is the most diverse dataset with images of dorsal fins. Some images contain only the dorsal fin, while others contain a significant part of the whaleś body.

IPanda50 [162] The authors collected giant pandas streaming videos from the Panda Channel, which contain daily routine videos of pandas of different ages. The identity

annotations are provided by professional zookeepers and breeders. The authors manually selected images with various illuminations, viewpoints, postures, and occlusions. In addition, they manually cropped out each individual panda with a tight bounding box and provided additional eye annotations.

LeopardID2022 [153] + **HyenaID2022** [153] HyenaID2022 and LeopardID2022 are datasets published by WildMe, representing a collaborative effort with the Botswana Predator Conservation Trust. These datasets include labeled images of hyenas and leopards collected for data collection and population modeling. The images include both daytime and nighttime photos of varying quality, with some making it challenging to spot the animals. Both datasets are annotated with viewpoints and bounding boxes.

LionData [50] + **NyalaData** [50] LionData and NyalaData contain images of lions and nyalas collected from the Mara Masia project in Kenya. While images in NyalaData are relatively uniform and show the image of the whole nyalas, LionData depicts various lion details such as ears or noses.

MacaqueFaces [166] MacaqueFaces shows the faces of group-housed rhesus macaques at a breeding facility in large indoor enclosures. To allow the care staff to identify individuals, the animals at the colony, the monkeys, were tattooed with an abbreviation of their ID on their chests. High-definition video footage was collected. Each video was annotated with the date and group information. Faces were semi-automatically extracted from videos, and random frames were selected for each individual. Only adults were included, as the facial features of infants changed substantially over the one-year filming period.

MPDD [70] The MPDD dataset consists of 1657 full-body images of 192 dogs in various poses and perspectives, designed to simulate natural environments and video surveillance diversity. Images were curated to exclude low-quality, blurry, or obscured photos. Each dog has an average of 9 images, and the dataset aims to represent diverse postures and perspectives.

NDD20 [152] The Northumberland Dolphin Dataset 2020 (NDD20) is a challenging image dataset as it contains both above and underwater photos of two dolphin species taken between 2011 and 2018. The datasets contain images taken both above and below water. The underwater collection consists of 36 opportunistic surveys of the Farne Deeps. Above-water collection consists of 27 surveys along a stretch of the Northumberland coast. Above-water photographs were taken using a camera from the deck of a small rigid inflatable boat on days of fair weather and good sea conditions. Underwater images are screen grabs from high-definition video footage taken with cameras again under good sea conditions. Individuals in the above water images are identified using the structure of the dolphin's dorsal fin. Underwater images are less common but provide additional features for identification, such as general coloring, unique body markings, scarring, and patterns formed by injury or skin disease. The images contain multiple annotations, including dolphin species, and approximately 14% of above-water images contain segmentation masks for the dolphin fin.

PolarBearVidID [176] PolarBearVidID is a dataset of 13 individual polar bears from 6 German zoos. The photos are extracted from 1431 video sequences at 12.5 fps, totaling around 138 thousand images. Since the cameras were stationary, the background was cropped to prevent background overfitting.

SealID [116] SealID is a Saimaa ringed seal database from Lake Saimaa in Finland. The data were collected annually during the Saimaa ringed seal molting season from 2010 to 2019 by both ordinary digital cameras during boat surveys and game camera traps. The GPS coordinates, the observation times, and the numbers of the seals were noted. Seal images were matched by an expert using individually characteristic fur patterns. The dataset contains patches and standard images. Patches show small patterned body parts, which are sufficient for seal identification. Standard images are presented both as full images and their segmented version with seal only and black background.

SeaStarReID2023 [160] This dataset contains 1204 images of 39 individual Asterias rubens sea stars and 983 images of 56 individual Anthenea australiae sea stars. For the ASRU data set, images were taken on five distinct days. For the ANAU data set, images were taken in three locations (sunlight, shaded, and naturalistic exhibit) on the same day. The photos were taken in a water tank.

SeaTurtleID [2] SeaTurtleID is a novel large-scale dataset of Mediterranean loggerhead sea turtles. These turtles are well-suited to photo identification due to their unique scale patterns, which can be used to identify individual turtles and remain stable throughout their lives. These patterns are found on the lateral and dorsal sides of the turtle's head and differ between the left and right sides of the same turtle.

The dataset contains photographs continuously captured over 12 years, from 2010 to 2021. With 7774 photographs and 400 individuals, the dataset represents the most extensive publicly available dataset for sea turtle identification in the wild. The images are uncropped, with various backgrounds, and each has a time stamp of capture. Approximately 90% of photographs have a size of 5472×3648 pixels. The average photograph size is 5289×3546 pixels, while the head occupies an average of 639×551 pixels. The photographs were captured using three different cameras with various accessories and taken from various distances at depths ranging from 1 to 8 meters, with most taken at less than 5 meters deep.

The annotation of individual identities for the SeaTurtleID dataset was done manually by an experienced curator and validated by automatic re-identification methods. Head segmentation masks and corresponding bounding boxes were generated using a combination of manual and machine annotation.

SMALST [178] SMALST is a unique dataset because it does not contain images of real animals. Instead, the authors used the SMALR method [177] to render 3D models of artificial Grevy's zebras from real zebra images. Then, they used projections to generate multiple zebra images from each 3D model. Finally, they put the generated image on some background images. The advantage of this approach is the possibility of generating an infinite number of images and having precise segmentations for free. The disadvantage is that the images are computer-generated and placed in a non-real background.

SouthernProvinceTurtles [1] SouthernProvinceTurtles is a collection of photos of green sea turtles from two different sources in the Southern Province of Sri Lanka. The first one comes from nesting turtles on the beaches of Rekawa and Batigama, and the second one from multiple rescue centers on the southern coast of Sri Lanka. There is a significant shift between those two. The first source contains generally difficult photos taken with the red light at night, while the second source contains standard photos.

StripeSpotter [89] StripeSpotter is the first published dataset. For seven consecutive days, the authors made a semi-random circuit through the 90,000-acre nature conservancy Ol'Pejeta Conservancy in Laikipia, which contains several hundred wild Plains zebras and fewer than 20 endangered Grevy's zebras. Two people were stationed on top of the vehicle to take pictures with cheap digital cameras while the driver circled around individual groups of zebras so as to capture both flanks of the animal. We collected as many pictures as possible of each flank of an animal in different positions in its natural walking gait. A professionally trained field assistant identified the images based on a database of prior sightings stretching back almost ten years. All but a few zebras were reliably identified.

WhaleSharkID [72] WhaleSharkID contains images of whale sharks and represents a collaborative effort based on the data collection and population modeling efforts conducted at Ningaloo Marine Park in Western Australia from 1995 to 2008. Images are annotated with bounding boxes around each visible whale shark and viewpoints.

ZindiTurtleRecall [155] ZindiTurtleRecall was collected through the Watamu Turtle Watch and Local Ocean Conservation. Many of the turtles in this project are turtles who have been caught as bycatch by fishermen and bought by Local Ocean Conservation for rehabilitation. Each rescued turtle is assessed, then measured, weighed, and tagged. If it is in good health, the turtle is transported to the Watamu Marine National Park, where it is released back into the ocean. Severely injured turtles are admitted to a rehabilitation unit. The dataset contains close-up images of turtle eyes and post-ocular scutes from three different viewpoints.

Chapter 5

MegaDescriptor

In this chapter, we introduce MegaDescriptor, a foundation model for animal identification based on the Swin Transformer architecture [98]. MegaDescriptor is trained using metric learning on a large dataset containing animal identities across various species. To support accessibility and ease of integration, we provide a variety of pre-trained MegaDescriptor models (Small, Medium, and Large) on a HuggingFace hub.

MegaDescriptor aims to generalize across species, addressing a common limitation of existing deep-learning models for animal identification. Most models are fine-tuned on single-species datasets and require species-specific adaptation, which limits their broader applicability. In contrast, large-scale foundational models like CLIP[129] and DINOv2[119] offer strong general vision capabilities but are not optimized for fine-grained individual animal identification. Distinguishing subtle differences in patterns and markings is crucial for this task. MegaDescriptor bridges this gap by achieving state-of-the-art performance on multiple animal identification datasets. It significantly outperforms existing pre-trained models, including CLIP and DINOv2.

The main contributions of this chapter are:

- An extensive experimental evaluation comparing existing animal identification methods and datasets.
- The introduction of MegaDescriptor, a foundational model for individual animal identification across multiple species.
- Ablation studies on the hyperparameters and training configurations of MegaDescriptor.

5.1 Methodology

We follow the closed-set setting for animal identification as described in Section 2.3, where the task is to assign identities from a predetermined set of known identities to given unseen images. In the search for the best suitable methods for the MegaDescriptor, we follow up on existing literature [125, 94, 45, 107] and focus on local descriptors and metric learning. We evaluate all the ablation studies over 29 datasets that were available at the time of training. The datasets are a subset of the datasets introduced in Section 4.2.

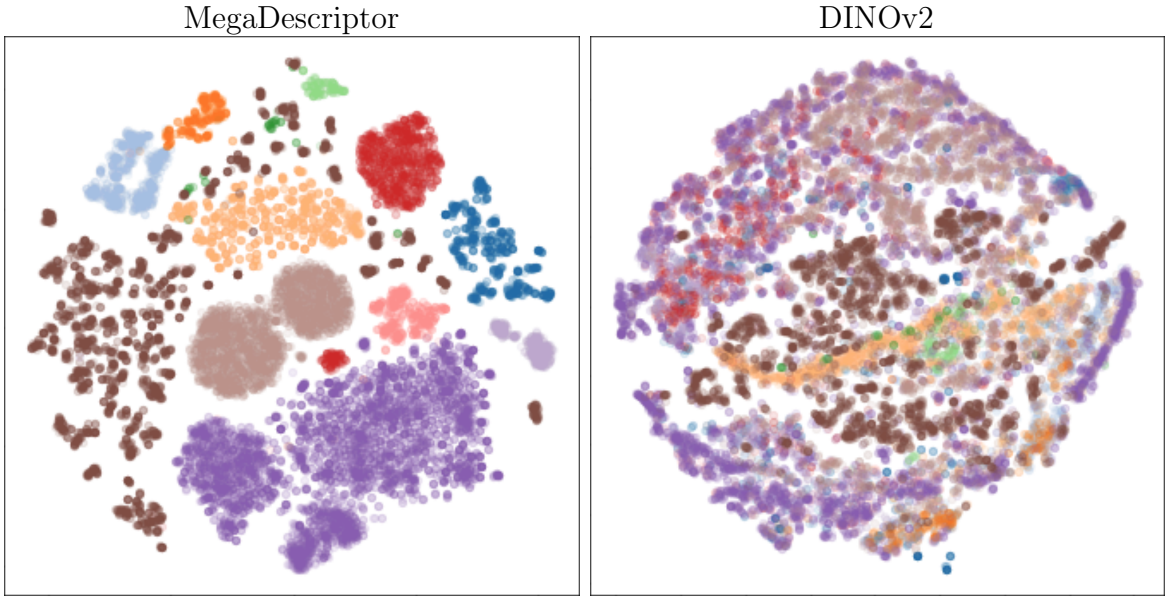


Figure 5.1: **Latent space separability of MegaDescriptor**. Embedding visualization (t-sne) of unseen individual animals (identity-wise) for the proposed MegaDescriptor and DINOv2. Colors represent different datasets (i.e., species).

5.1.1 Local features approaches

Drawing inspiration from the success of local descriptors in existing animal identification tools [53, 125], we include the SIFT and Superpoint descriptors in our evaluation. The matching process includes the following steps: (i) we extract keypoints and their corresponding descriptors from all images in reference and query sets, (ii) we compute the descriptors distance between all possible pairs of reference and query images, (iii) we employ a ratio test with a threshold to eliminate potentially false matches, with the optimal threshold values determined by matching performance on the reference set, and (iv) we determine the identity based on the absolute number of correspondences, predicting the identity with highest number from reference set.

5.1.2 Metric learning approaches

Following the recent progress in human and vehicle re-id[38, 168, 115], we select two metric learning methods for our ablation studies – Arcface [47] and Triplet loss [144] – which both learn a representation function that maps objects into deep embedding space. More details about metric learning methods are in Section 2.4.

Matching strategy In the context of our extensive experimental scope, we adopt a simplified approach to determine the identity of query (i.e., test) images, relying solely on the closest match within the reference set. To frame this in machine learning terminology, we essentially create a 1-nearest-neighbor classifier within a deep-embedding space using cosine similarity.

Training strategy While training models, we use all 29 publicly available datasets provided through the WildlifeDataset toolkit. All datasets were split in an 80/20% ratio for reference and query sets, respectively, while preserving the closed set setting, i.e., all

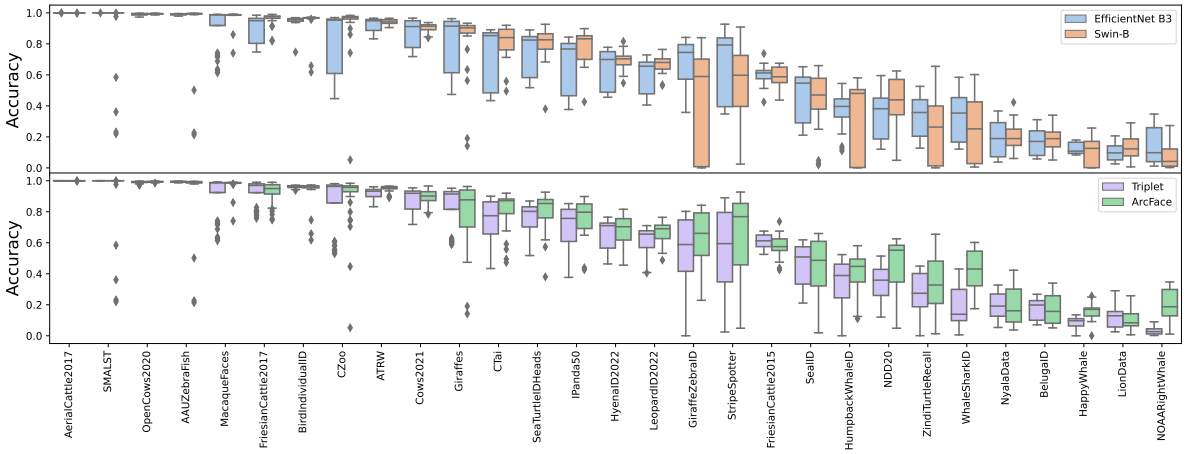


Figure 5.2: **Ablation of the backbone architecture and metric learning method**. We compare two backbones – Swin-B and EfficientNet-B3 – and Triplet / ArcFace methods on all available animal identification datasets. In most cases, the Swin-B with ArcFace maintains competitive or better performance than EfficientNet-B3 and Triplet.

identities in the query set are available in the reference set. Models were optimized using the SGD optimizer with momentum (0.9) for 100 epochs using the cosine annealing learning rate schedule and mini-batch of 128.

Hyperparameter tunning The performance of the metric learning approaches is usually highly dependent on training data and optimization hyperparameters [115]. Therefore, we perform an exhaustive hyperparameter search to determine optimal hyperparameters with sustainable performance in all potential scenarios and datasets for both methods. Besides, we compare two backbone architectures – EfficientNet-B3 and Swin-B – with a comparable number of parameters. We select EfficientNet-B3 as a representative of traditional convolutional-based and Swin-B as a novel transformer-based architecture.

For each architecture type and metric learning approach, we run a grid search over selected hyperparameters and all the datasets. We consider 72 different settings for each dataset, yielding 2088 training runs. We use the same optimization strategy as described above. All relevant hyperparameters and their appropriate values are listed in Table 5.1.

Backbone	$\{\texttt{Swin}-\texttt{B}, \texttt{EfficientNet}-\texttt{B3}\}$
Learning rate	$\{0.01, 0.001\}$
ArcFace margin	$\{0.25, 0.5, 0.75\}$
ArcFace scale	$\{32, 64, 128\}$
Triplet mining	$\{\texttt{all}, \texttt{semi}, \texttt{hard}\}$
Triplet margin	$\{0.1, 0.2, 0.3\}$

Table 5.1: **Grid-search setup**. Selected hyperparameters and their appropriate values for ArcFace and Triplet approaches.

5.2 Ablation studies

This section presents a set of ablation studies to empirically validate the design choices related to model distillation (i.e., selecting methods, architectures, and appropriate hyperparameters) while constructing the MegaDescriptor feature extractor, i.e., the first-ever foundation model for animal identification. Furthermore, we provide both qualitative and quantitative performance evaluation comparing the newly proposed MegaDescriptor in a zero-shot setting with other methods, including SIFT, Superpoint, ImageNet, CLIP, and DINOv2.

5.2.1 Loss and backbone components

To determine the optimal metric learning loss function and backbone architecture configuration, we conducted an ablation study, comparing the performance (median accuracy) of ArcFace and Triplet loss with either a transformer- (Swin-B) or CNN-based backbone (EfficientNet-B3) on all available identification datasets. In most cases, the Swin-B with ArcFace combination maintains competitive or better performance than other variants. Overall, ArcFace and transformer-based backbone (Swin-B) performed better than Triplet and CNN backbone (EfficientNet-B3). First quantiles and top whiskers indicate that Triplet loss underperforms compared to ArcFace even with correctly set hyperparameters. The full comparison in the form of a box plot is provided in Figure 5.2.

5.2.2 Hyperparameter tunning

In order to overcome the performance sensitivity of metric learning approaches regarding hyperparameter selection and to select the generally optimal parameters, we have performed a comprehensive grid search strategy.

Following the results from the previous ablation, we evaluate how various hyperparameter settings affect the performance of a Swin-B backbone optimized with Arcface and Triplet losses. In the case of ArcFace, the best setting (i.e., $lr = 0.001$, $m = 0.5$, and $s = 64$) achieved a median performance of 87.3% with 25% and 75% quantiles of 49.2% and 96.4%, respectively. Interestingly, three settings underperformed by a significant margin, most likely due to unexpected divergence in the training[1]. The worst settings achieved a mean accuracy of 6.4%, 6.1%, and 4.0%. Compared to ArcFace, Triplet loss configurations showed higher performance on both 25% and 75% quantiles, indicating significant performance variability.

The outcomes of the study are visualized in Figure 5.3 as a boxplot, where each box consists of 29 values.

[1]These three settings were excluded from further evaluation and visualization for a more fair comparison.

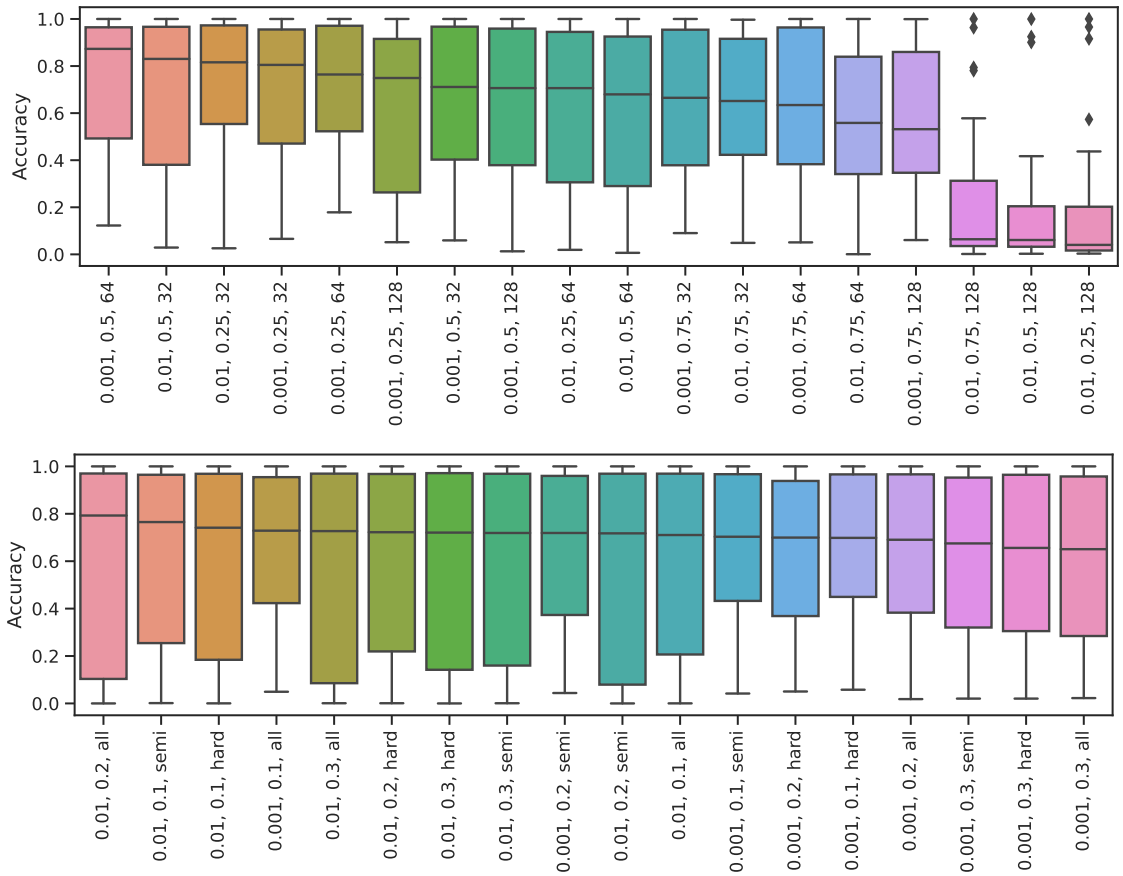


Figure 5.3: **Ablation of hyperparameters search**. We display performance for all settings as a boxplot combining accuracy from all 29 datasets. ArcFace (Top) and Triplet loss (Bottom).

5.2.3 Metric learning vs. Local features

The results conducted over 29 datasets suggested that both metric learning approaches (Triplet and ArcFace) outperformed the local-feature-based methods on most datasets by a significant margin. The comparison of local-feature-based methods (SIFT and Superpoint) revealed that Superpoints are a better fit for animal identification, even though they are rarely used over SIFT descriptors in the literature. A detailed comparison is provided in Table 5.2. Note that the Giraffes dataset was labeled using local descriptors; hence, the performance is inflated and better than for metric learning.

The same experiment revealed that several datasets, e.g., AerialCattle2017, SMALST, MacaqueFaces, Giraffes, and AAUZebraFish, are solved or close to that point and should be omitted from development and benchmarking.

5.3 Performance evaluation

Insights from our ablation studies led to the creation of MegaDescriptors – the Swin-transformer-based models optimized with ArcFace loss and optimal hyperparameters using all publicly available animal identification datasets.

In order to verify the expected outcomes, we perform a similar comparison as in metric learning vs. Local features ablation, and we compare the MegaDescriptor with CLIP (ViT-L/p14-336), ImageNet-1k (Swin-B/p4-w7-224), and DINOv2 (ViT-L/p14-518) pre-

Dataset	SIFT	Superpoint	Triplet	ArcFace
AAUZebraFish	65.09	25.09	99.40	98.95
ATRW	89.30	92.74	93.26	95.63
AerialCattle2017	98.96	99.06	100.0	100.0
BelugaID	1.10	0.02	19.85	15.74
BirdIndividualID	48.96	48.71	96.45	96.00
CTai	33.87	29.58	77.44	87.14
CZoo	67.61	83.92	96.34	95.75
Cows2021	58.82	75.89	91.90	90.14
FriesianCattle2015	56.25	55.00	61.25	57.50
FriesianCattle2017	85.86	86.87	96.97	94.95
GiraffeZebraID	74.45	73.85	58.85	66.07
Giraffes	97.01	99.25	91.42	88.69
HappyWhale	0.38	0.42	9.73	17.03
HumpbackWhaleID	11.65	11.82	38.78	44.75
HyenaID2022	39.84	46.67	71.03	70.32
IPanda50	35.12	47.35	75.71	79.71
LeopardID2022	72.71	75.08	65.56	69.02
LionData	31.61	5.16	12.90	8.39
MacaqueFaces	75.72	75.08	98.69	98.73
NDD20	17.14	29.01	35.88	55.18
NOAARightWhale	6.53	15.31	2.68	18.74
NyalaData	10.75	18.46	19.16	19.85
OpenCows2020	72.76	86.38	99.31	99.37
SMALST	92.22	98.37	100.0	100.0
SeaTurtleIDHeads	55.23	80.58	80.22	85.32
SealID	31.41	62.11	50.84	48.68
StripeSpotter	70.12	94.51	59.45	76.83
WhaleSharkID	4.29	22.90	13.88	43.10
ZindiTurtleRecall	17.91	25.73	27.40	32.74

Table 5.2: **Ablation of animal identification methods**. We compare two local-feature (SIFT and Superpoint) methods with two metric learning approaches (Triplet and ArcFace). Metric learning approaches outperformed the local-feature methods on most datasets. ArcFace provides more consistent performance. For metric learning, we list the median from the previous ablation.

trained models. The proposed MegaDescriptor with Swin-L/p4-w12-384 backbone performs consistently on all datasets and outperforms all methods on all 29 datasets. Notably, the state-of-the-art foundation model for almost any vision task – DINOv2 – with a much higher input size (518×518) and larger backbone performs poorly in animal identification.

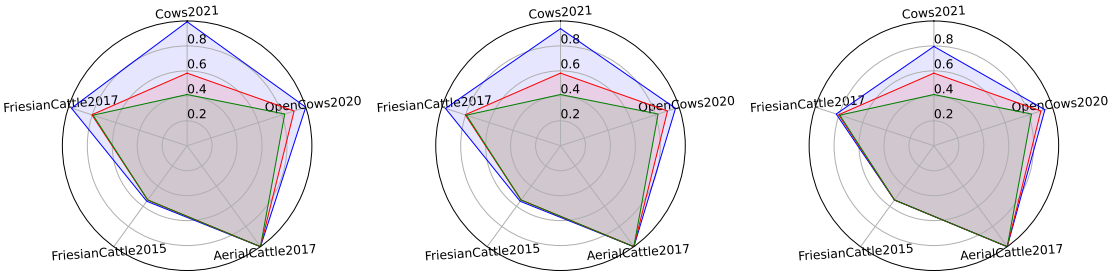
5.3.1 Seen and unseen domain performance

This section illustrates how the proposed MegaDescriptor can effectively leverage features learned from different datasets and its ability to generalize beyond the datasets it was initially fine-tuned on. By performing this experiment, we try to mimic how the MegaDescriptor will perform on *Seen (known)* and *Unseen Domains (unknown)*.

We evaluate the generalization capabilities using the MegaDescriptor-B and all available datasets from one domain (cattle), e.g., AerialCattle2017, FriesianCattle2015, FriesianCattle2017, Cows2021, and OpenCows2020. The first mutation (*Same Dataset*) was trained on training data from all datasets and evaluated on test data. The second mutation (*Seen Domain*) used just the part of the domain for training; OpenCows2020 and

Dataset	ImageNet	CLIP	DINOv2	MegaDesc.
AAUZebraFish	94.38	94.91	96.93	99.93
ATRW	88.37	86.88	88.47	94.33
AerialCattle2017	100.0	99.99	100.0	100.0
BelugaID	19.58	11.20	14.64	66.48
BirdIndividualID	63.11	52.75	74.90	97.82
CTai	60.99	50.38	68.70	91.10
CZoo	78.49	58.87	87.00	99.05
Cows2021	57.84	41.06	58.19	99.54
FriesianCattle2015	55.00	53.75	55.00	55.00
FriesianCattle2017	83.84	79.29	80.30	96.46
GiraffeZebraID	21.89	32.47	37.99	83.17
Giraffes	59.70	42.16	60.82	91.04
HappyWhale	14.25	15.30	13.26	34.30
HumpbackWhaleID	7.32	3.23	6.44	77.81
HyenaID2022	46.83	45.71	49.52	78.41
IPanda50	72.51	57.60	62.84	86.91
LeopardID2022	61.13	59.94	57.50	75.58
LionData	20.65	5.16	12.90	25.16
MacaqueFaces	78.58	64.17	91.56	99.04
NDD20	43.13	46.70	37.85	67.42
NOAARightWhale	28.37	28.27	24.84	40.26
NyalaData	10.28	10.51	14.72	36.45
OpenCows2020	92.29	82.26	90.18	100.0
SMALST	91.25	83.04	94.63	100.0
SeaTurtleIDHeads	43.84	33.57	46.08	91.18
SealID	41.73	34.05	29.26	78.66
StripeSpotter	73.17	66.46	82.93	98.17
WhaleSharkID	28.26	26.37	22.02	62.02
ZindiTurtleRecall	15.61	12.26	14.83	74.40

Table 5.3: **Animal identification performance**. We compare the MegaDescriptor-L (Swin-L/p4-w12-384) among available pre-trained models, e.g., ImageNet-1k (Swin-B/p4-w7-224), CLIP (ViT-L/p14-336), and DINOv2 (ViT-L/p14-518). The proposed MegaDescriptor-L provides consistent performance on all datasets and outperforms all methods on all 29 datasets.



Same Dataset *Seen Domain* *Unseen Domain*

Figure 5.4: **Seen domain and un-seen domain performance**. We compare the performance of a MegaDescriptor-B (Swin-B/p4-w7-224), CLIP (ViT-L/p14-336) and DINOv2 (ViT-L/p14-518) on (i) *Same Dataset*: all datasets were used for fine-tuning, (ii) *Seen Domain*: Cows 2021 and OpenCows2020 were not used for fine-tuning, and (iii) *Unseen Domain*: no datasets were used for fine-tuning.

Cows2021 datasets were excluded. The third mutation (*Unseen Domain*) excludes all the cattle datasets from training.

The MegaDescriptor-B, compared with a CLIP and DINOv2, yields significantly better or competitive performance (see Figure 5.4). This can be attributed to the capacity of MegaDescriptor to exploit not just cattle-specific features. Upon excluding two cattle datasets (OpenCows2020 and Cows2021) from the training set, the MegaDescriptor's performance on those two datasets slightly decreases but still performs significantly better than DINOv2. This observation serves as evidence that the MegaDescriptor effectively exploits the features acquired from the remaining cattle datasets and other animal datasets. The MegaDescriptor retains reasonable performance on the cattle datasets even when removing cattle images from training. We attribute this to learning general fine-grained features, which is essential for all the identification in any animal datasets, and subsequently transferring this knowledge to the identification of the cattle.

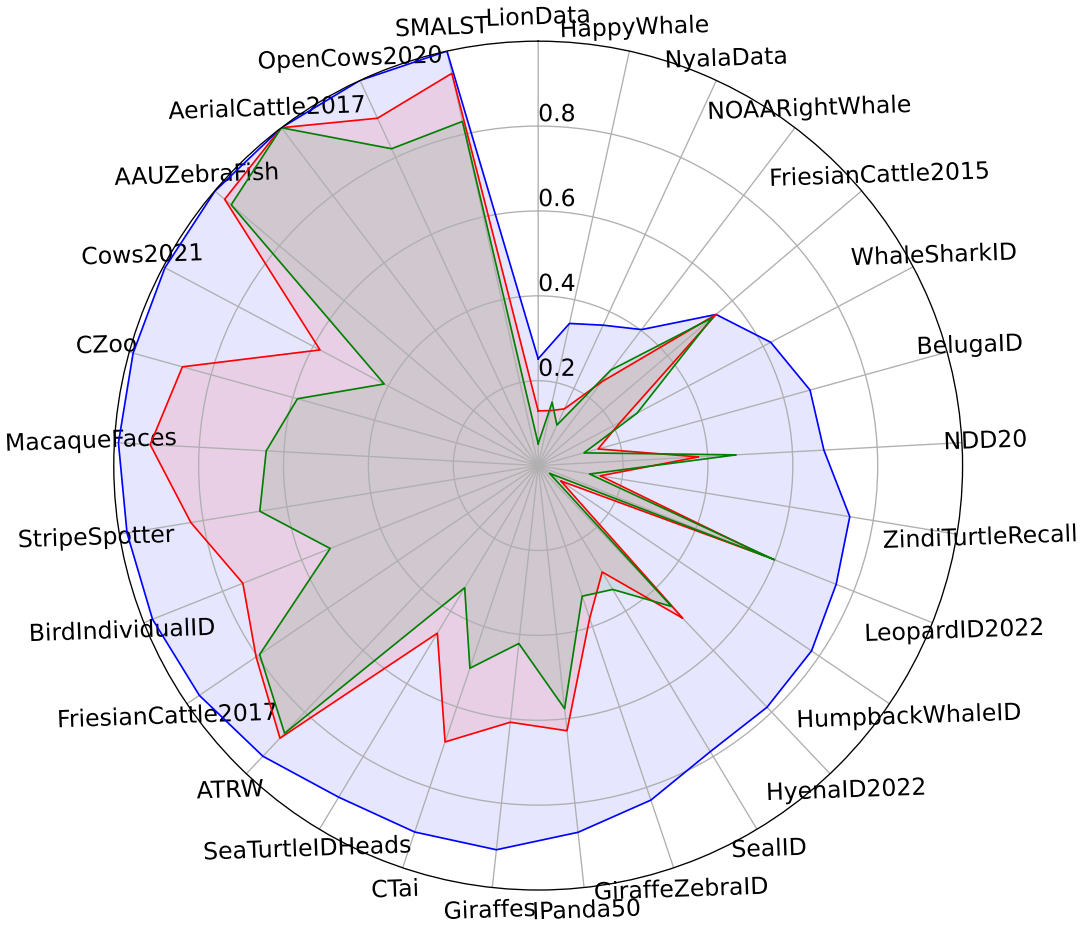


Figure 5.5: ***Pre-trained* models performance evaluation.** We compare DINOv2 (ViT-L/p14-518), CLIP (ViT-L/p14-336), and MegaDescriptor-L (Swin-L/p4-w12-384) on 29 selected datasets.

5.3.2 Effect of model size

To showcase and quantify the performance of different MegaDescriptor flavors, we compare five variants, e.g., **B**ase, **S**mall, **T**iny, and **L**arge-224 and **L**arge-384, originating from corresponding variations of the Swin architecture. All the models were trained and evaluated using the same setting. Naturally, the model performance in terms of accuracy increased with an increasing model size, i.e., the MegaDescriptor-L-384 outperformed smaller flavors by a considerable margin in most cases. Overall, higher model complexity achieved higher performance with few exceptions, where it underperformed by a small margin, e.g., by 2.53%, 0.48%, and 0.08% on FriesianCattle2017, LeopardID2022, and MacaqueFaces respectively. This is more or less statistically insignificant, given the poor quality of the data and the data acquisition.

We visualized the accuracy of all provided MegaDescriptor flavors in Figure 5.6 and Table 5.4.

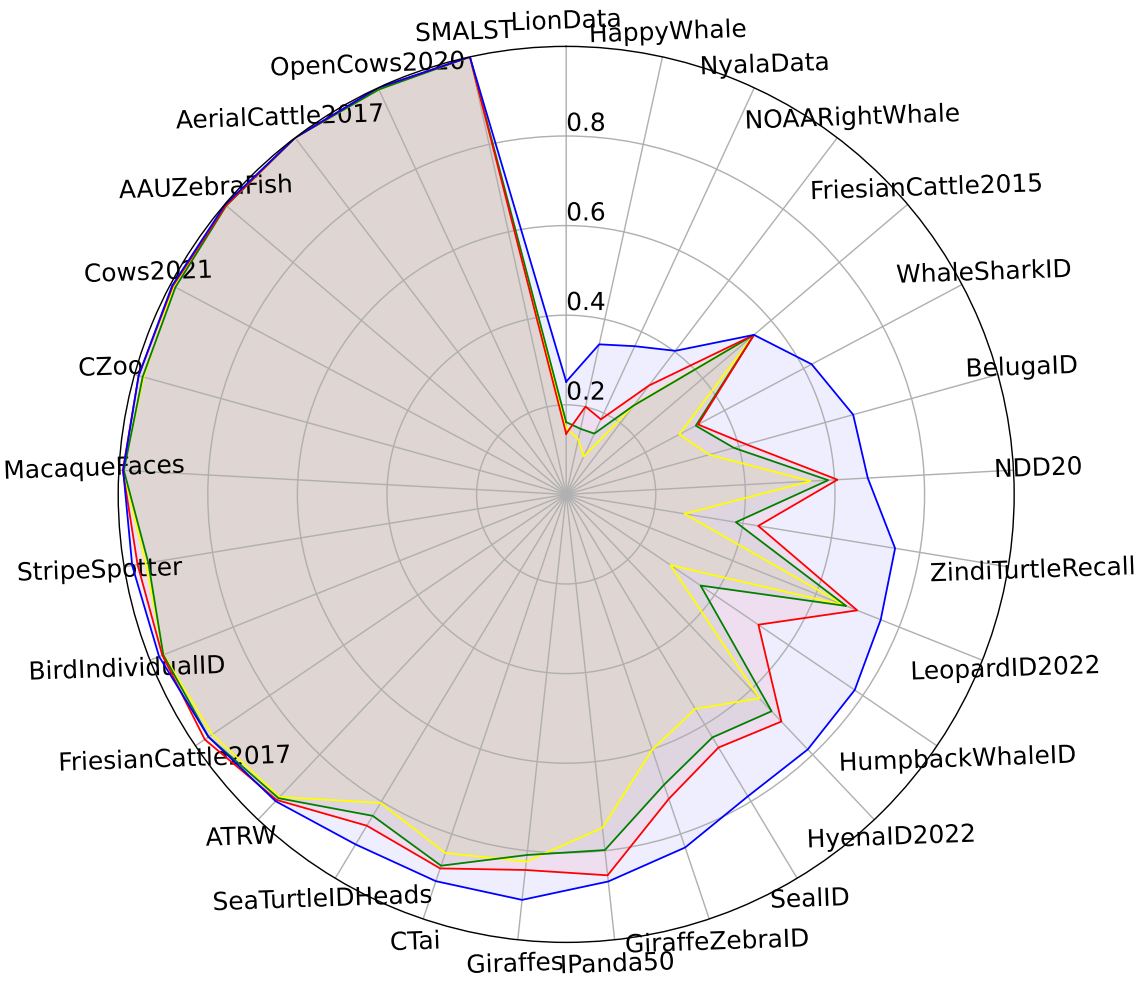


Figure 5.6: **Ablation study on model size/complexity.** Accuracy of different MegaDescriptor flavors MegaDescriptor-L (Swin-L/p4-w12-384), MegaDescriptor-B (Swin-B/p4-w7-224), MegaDescriptor-S (Swin-S/p4-w7-224), and MegaDescriptor-T (Swin-T/p4-w7-224) on 29 animal identification datasets.

	MegaDescriptor-T-224	MegaDescriptor-S-224	MegaDescriptor-B-224	MegaDescriptor-L-224	MegaDescriptor-L-384
AAUZebraFish	99.55	99.55	99.63	99.85	99.93
ATRW	93.02	93.40	93.95	93.67	94.33
AerialCattle2017	100.0	100.0	100.0	100.0	100.0
BelugaID	33.12	38.84	41.74	47.92	66.48
BirdIndividualID	96.73	96.81	97.04	97.21	97.82
CTai	84.46	87.46	88.10	90.68	91.10
CZoo	97.87	98.11	99.05	98.35	99.05
Cows2021	99.13	98.73	99.37	99.37	99.54
FriesianCattle2015	55.00	55.00	55.00	55.00	55.00
FriesianCattle2017	95.45	96.46	97.47	98.99	96.46
GiraffeZebraID	60.15	68.40	71.72	78.04	83.17
Giraffes	82.46	80.97	84.33	87.69	91.04
HappyWhale	12.58	14.98	20.07	25.34	34.30
HumpbackWhaleID	28.12	36.25	51.83	63.54	77.81
HyenaID2022	62.70	66.67	69.84	77.30	78.41
IPanda50	74.84	79.85	85.53	85.45	86.91
LeopardID2022	67.06	67.27	69.92	76.06	75.58
LionData	14.84	16.13	13.55	20.65	25.16
MacaqueFaces	99.04	98.89	99.12	98.96	99.04
NDD20	54.61	58.57	60.64	61.58	67.42
NOAARightWhale	25.16	24.95	30.51	34.69	40.26
NyalaData	9.35	14.95	18.46	21.73	36.45
OpenCows2020	99.58	99.58	100.0	99.79	100.0
SMALST	100.0	100.0	100.0	100.0	100.0
SeaTurtleIDHeads	80.38	83.74	86.31	89.86	91.18
SealID	55.88	63.31	65.95	70.02	78.66
StripeSpotter	95.12	94.51	96.95	97.56	98.17
WhaleSharkID	28.58	32.74	33.31	50.03	62.02
ZindiTurtleRecall	26.77	38.38	43.45	58.14	74.40

Table 5.4: **Ablation study on model size/complexity**. We compare five MegaDescriptor flavors, e.g., **L**arge, **B**ase, **S**mall, and **T**iny, in terms of accuracy. In general, models with a bigger model size or higher input resolution outperformed their *smaller* variants by a considerable margin. The best-performing model – MegaDescriptor-L-384 – underperformed by 2.53%, 0.48%, and 0.08% on FriesianCattle2017, LeopardID2022, and MacaqueFaces, respectively.

Chapter 6

WildFusion

In this chapter, we introduce WildFusion, a new state-of-the-art approach for zero-shot animal identification. It fuses calibrated deep similarity functions (i.e., MegaDescriptor or DINOv2 feature similarity) and local matching scores (i.e., number of matches from descriptors such as LoFTR and LightGlue) to select an identity from a database. For reference, see the illustration in Figure 6.1. With this straightforward approach, WildFusion significantly outperforms the current state-of-the-art without domain adaptation or fine-tuning.

Our method is easy to use in real applications as it does not require training and is usable out of the box with any pre-trained deep embedding models and local feature-matching methods. Even though the best results were obtained with dataset-specific calibration, we have empirically shown that using WildFusion of only local similarity score and with generic calibration still gives good performance, with mean accuracy dropping only by 2.3% and still reduced the relative error of MegaDescriptor by 44 percentage points. WildFusion's flexibility was also further proven by its strong performance in zero-shot settings tested on species "never seen before." The scalability and generalization potential of WildFusion makes it suitable for application across different species and environments, contributing significantly to the field of animal identification.

The main contributions of this chapter are:

- A new ensembling framework (WildFusion) that allows a combination of deep- and local-feature matching scores.
- A state-of-the-art performance on a set of animal identification problems, outperforming current methods by 8.5% on average; measured on 17 datasets.
- Comprehensive evaluation of selected state-of-the-art deep-learning and local feature-matching methods for image-matching and animal identification.
- Showing that WildFusion works without the need for fine-tuning and provides state-of-the-art performance out of the box, even in a zero-shot setting.

6.1 Methodology

In this section, we describe similarity scores based on deep embeddings and local feature matching and introduce combined WildFusion score. Our setting corresponds to the image retrieval approach to animal identification, as described in Section 2.3.2. The goal is to

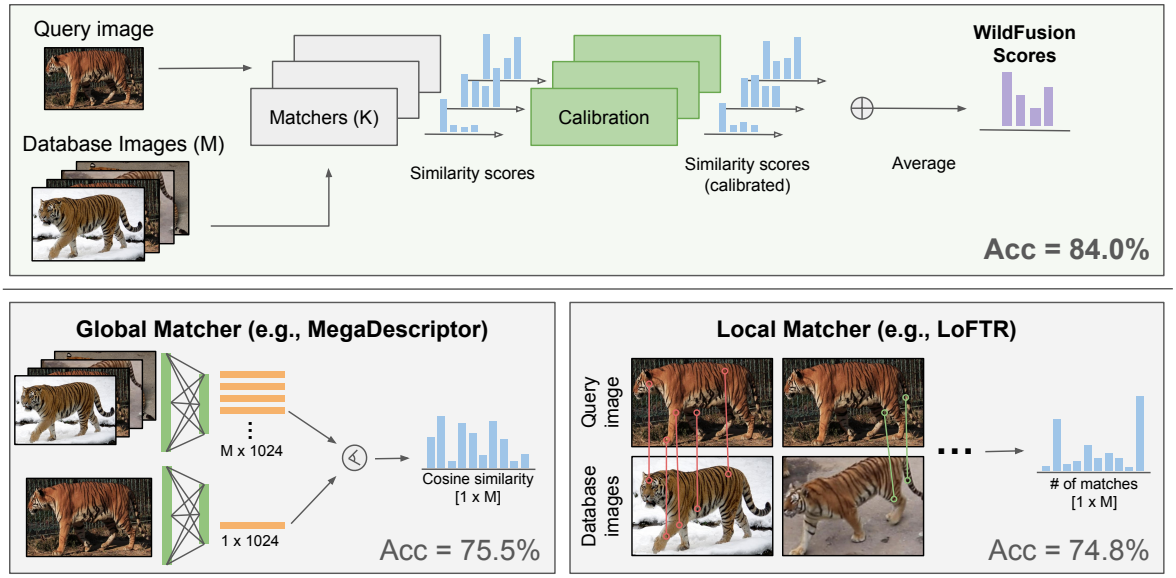


Figure 6.1: **Calibrated similarity fusion.** Fusing local (in the $[0, \mathcal{R}]$ range) and global matching scores (e.g., cosine similarity) is not possible without calibration. By calibrating the outputs of any local and global matcher, we can easily fuse them and achieve better performance. Across 17 datasets, we achieved an average performance improvement of 8.5% and a 35% reduction in relative error.

find the most visually similar images (whose identity is used as prediction) in the database of images $x_j, \ldots, x_D$ for a given query image x_q based on a similarity score.

6.1.1 Global similarity score

Given an image x, we use a neural network $f(x)$ to extract a fixed-length embedding. The network $f(x)$ is a complex function that maps images into embedding space where the representations of images depicting the same animal are closer together, while those of different individual animals are distinctively separated. Common architectures of neural networks include convolutional[99, 167] or transformer-based[51, 98] architectures and are often trained with metric learning, e.g., ArcFace[47] and Triplet loss[144], to promote separability in the embedding space. The similarity between images is calculated as the similarity between their representation in the embedding space. The details of metric learning algorithms are discussed in Section 2.4. Formally, we define the *global similarity* between two images x_i and x_j as the cosine similarity between their corresponding deep embeddings extracted by neural network f:

$$s_G(x_i, x_j) = \frac{f(x_i) \cdot f(x_j)}{\|f(x_i)\| \|f(x_j)|}. \tag{6.1}$$

6.1.2 Matching-based similarity score

We derive a similarity metric based on local feature matching as the number of found significant matches. The feature matching methods return a list of potential matches and their confidence score. We declare a match to be significant if its confidence score is above some threshold μ. Formally, given two images x_i and x_j with the number of matches $M(x_i, x_j)$ with confidence scores $c_m(x_i, x_j)$, we define the *local similarity* metric

as

$$s_L(x_i, x_j) = \sum_{m=1}^{M(x_i,x_j)} I\big(c_m(x_i, x_j) > \mu\big), \tag{6.2}$$

where I is the counting (0/1) function. This approach requires the tuning of the thresholding hyperparameter μ, where large values of μ allow for a small number of high-quality matches, while low values of μ result in a larger amount of matches with potentially lower quality. As we empirically show in Section 6.3.1, all considered feature matching methods are robust to μ selections, with $\mu = 0.5$ being a reasonable choice in most scenarios.

6.1.3 Score calibration

Calibration refers to rescaling model outputs so that they can be interpreted probabilistically. In our case, normalizing the outputs of multiple models to the common range $[0, 1]$ is required. The predictions of a well-calibrated model reflect its confidence in the given class predictions [61].

We apply calibration to the predicted similarity scores. The similarity scores are used for comparison and ranking in image retrieval, with the magnitude of the scores having no direct interpretation. We use calibration to ensure that the predicted similarity score corresponds to the probability that the images in a pair have the same identity. We construct calibrated scores from either global or matching-based scores using a calibration function $f_{\text{cal}} : \mathbb{R} \to [0, 1]$.

$$\hat{s}(x_i, x_j) = f_{cal}(s(x_i, x_j)). \tag{6.3}$$

A common approach for constructing f_{cal} is Platt scaling[127], which involves fitting a single-variable logistic regression with uncalibrated scores as inputs. Another widely used method is isotonic regression[172], a variant of binning regression with a monotonicity constraint. Given uncalibrated scores, it learns a non-decreasing piecewise constant function. However, for our application in ranking and image retrieval, we require a strictly increasing function to handle ties in scores. To achieve this, we first apply the isotonic regression and second interpolate the bin centers using a Piecewise Cubic Hermite Interpolating Polynomial (PCHIP)[59], which performs cubic interpolation while preserving monotonicity. This procedure results in the required strictly increasing calibration function.

6.1.4 WildFusion – Calibrated score ensembling

To construct an ensemble, we consider K models with similarity scores $s_i(x_i, x_j)$ for a pair of images (x_i, x_j). The calibrated scores $\hat{s}_i(x_i, x_j)$ are interpreted as estimates of same true probability $P(\text{id}(x_i) = \text{id}(x_j) \mid x_i, x_j)$. To denoise the prediction, we assume the probabilities to be independent observations with additive noise. The WildFusion score is thus a weighted average of n calibrated scores:

$$s_F(x_i, x_j) = \sum_{i=1}^{K} w_i \hat{s}_i(x_i, x_j), \tag{6.4}$$

where the weights should reflect the variance of the additive noise in the score. If we assume that all scores have similar variance, equal weights $w_i = \frac{1}{n}$ are selected, and the weighted average reduces to the simple average.

Figure 6.2: **Distinct animal features for identification**. Based on the natural visual appearance, the most distinguishable features for animals are spots, stripes, facial landmarks, and the shape of body parts (e.g., ears for elephants and fin for whales).

6.2 Experiments

6.2.1 Datasets

We evaluate WildFusion on 17 datasets[1] that include diverse species of animals. The datasets were acquired with the help of the library Wildlife Datasets introduced in Chapter 4. To make our experiments more feasible, we excluded datasets that are saturated in performance or have a large number of images. To construct the appropriate training (*database*) and test (*query*) datasets, we follow the same methodology as in Chapter 5. In addition, we corrected inconsistencies caused by the incorrect loading of images for ATRW and NDD20 datasets due to multiple identities in one image. For these datasets, we fixed the loading by applying the appropriate bounding box or segmentation mask. Therefore, as loaded images are not exactly the same, the results achieved for these two datasets are higher than those in the original work. Sample images from selected datasets are shown in Figure 6.2. For basic statistics of the datasets, see Table 6.1.

Evaluation protocol We consider the same closed-set splits as in Chapter 5, meaning that all individual animals are both in the database (training) and query (test) sets. We approach the problem as image retrieval: for each image in the query set, we find an image in the database and make the query prediction have the same identity as the image from the database. Performance in all experiments is measured as top-1 accuracy.

WildFusion relies on finding hyper-parameters and fitting calibration models. The standard approach involves splitting the training set into development and validation parts, which are used for the selection of the best hyper-parameters and fitting calibration models. However, this approach is not applicable in our case because the MegaDescriptor was trained on the whole training set and cannot be used for validation. We addressed this issue by splitting the original test set into a validation set and a new, smaller test set using a 0.5 ratio. We estimated both μ and the calibration function on the validation set and utilized them for the final prediction on the test set. Due to this change in the test set, our results are not directly comparable to the results reported by[31].

Technical details To construct the global scores, we use embeddings extracted by MegaDescriptor[31] and DINOv2[119]. For local matching scores, we use LightGlue[96]

[1]For zero-shot, we use only two that were not used in the MegaDescriptor training.

Table 6.1: **Characteristics of datasets used in experiments**. †Used in zero-shot scenario.

	category	# of images	# of individuals
ATRW[94]	tigers	5,415	182
CowDataset† [60]	cows	1485	13
GiraffeZebraID[123]	giraffes, zebras	6,925	2,056
Giraffes[107]	giraffes	1,393	178
HyenaID2022[153]	hyenas	3,129	256
LeopardID2022[153]	leopards	6,806	430
NyalaData[50]	nyalas	1,942	237
SealID[116]	seals	2,080	57
SeaStarReID2023† [160]	starfish	2187	95
SeaTurtleID[2]	sea turtles	7,774	400
WhaleSharkID[72]	whale sharks	7,693	543
ZindiTurtleRecall[155]	sea turtles	12,803	2,265
BelugaID[13]	belugas	5,902	788
CTai[57]	chimpanzees	4,662	71
IPanda50[161]	pandas	6,874	50
NDD20[152]	dolphins	2,657	82
NOAARightWhale[133]	whales	4,544	447

feature matching with local descriptors ALIKED[174], DISK[157], and SuperPoint[49]. We use at most 512 keypoints, and their appropriate descriptors, extracted from images resized to 512×512. For matching with LoFTR[147], we use the outdoor variant trained on the MegaDepth[95] dataset. On input, we use image pairs with both images resized to 512×512. A total of four local feature matching methods were considered to construct matching-based scores. All these methods were taken off-the-shelf, and none were fine-tuned or retrained. We perform the experiments on the datasets described in Section 6.2.1. In the baseline WildFusion, we search for optimal hyperparameter μ from Equation 6.2 separately for each dataset. The calibrated scores are given equal weights w_i. The summary of settings is in Table 6.2.

Table 6.2: **WildFusion settings overview**. We test a variety of state-of-the-art local and global methods for animal identification and image retrieval. The calibration is done using Logistic or Isotonic regression.

Components:	Local matching methods: – LoFTR – LightGlue + SuperPoint – LightGlue + Disk – LightGlue + Aliked	Global similarity methods: – MegaDescriptor-L-384 – DINOv2-512
Calibration:	Isotonic regression with PCHIP interpolation Logistic regression	
Fusion:	Average with equal weights w_i	

6.2.2 Baseline Performance

WildFusion clearly outperforms MegaDescriptor in most scenarios. When using all available scores, WildFusion shows superior performance on 16 out of 17 datasets, with only one dataset, ZindiTurtleRecall, showing a slight decrease in accuracy. The average accuracy improvement is substantial, with WildFusion (all) achieving 84.0% compared to MegaDescriptor's 75.5%, representing a notable average gain of 8.5 percentage points. The most significant improvements are seen in datasets like NDD20, WhaleSharkID, SeaStarReID2023, and SealID, where WildFusion shows accuracy gains of over 14 percentage points.

Interestingly, even when using only local matching scores, WildFusion maintains competitive performance. It outperforms MegaDescriptor on 11 out of 17 datasets with average accuracy (78.5%), which is better than MegaDescriptor by 3.0 percentage points. This suggests that the local matching scores are quite powerful on their own, without any need for fine-tuning on animal datasets. More details about the results, including per dataset performance, are in Table 6.3. Besides, we provide a qualitative evaluation in Figure 6.3

Table 6.3: **WildFusion's performance in comparison with MegaDescriptor**. On average, WildFusion, outperforms MegaDescriptor, even with just *local* descriptors. WildFusion with *all* local and deep descriptors ranks the best on all but two datasets.

	MegaDescriptor Large-384	(*all*) WildFusion	Δ	(*local*) WildFusion	Δ
ZindiTurtleRecall	74.24	71.90	-2.34	45.62	-28.62
CTai	91.86	92.08	+0.21	81.80	-10.06
ATRW	97.96	98.51	+0.56	98.33	+0.37
CowDataset	98.66	100.00	+1.34	100.00	+1.34
SeaTurtleIDHeads	91.18	95.00	+3.82	93.82	+2.63
IPanda50	85.76	89.68	+3.92	81.40	-4.36
NyalaData	41.59	46.26	+4.67	25.23	-16.36
BelugaID	67.61	72.46	+4.85	63.07	-4.54
NOAARightWhale	43.25	49.25	+6.00	42.18	-1.07
Giraffes	91.04	99.25	+8.21	98.51	+7.46
HyenaID2022	78.41	90.48	+12.06	88.25	+9.84
GiraffeZebraID	82.98	95.74	+12.77	94.81	+11.84
LeopardID2022	77.82	90.93	+13.11	89.40	+11.58
SealID	78.47	92.82	+14.35	90.91	+12.44
SeaStarReID2023	82.24	99.53	+17.29	100.00	+17.76
WhaleSharkID	62.04	80.33	+18.28	77.68	+15.64
NDD20	38.35	63.53	+25.19	63.16	+24.81
Average	*75.50*	*83.99*	*+8.49*	*78.48*	*+2.98*

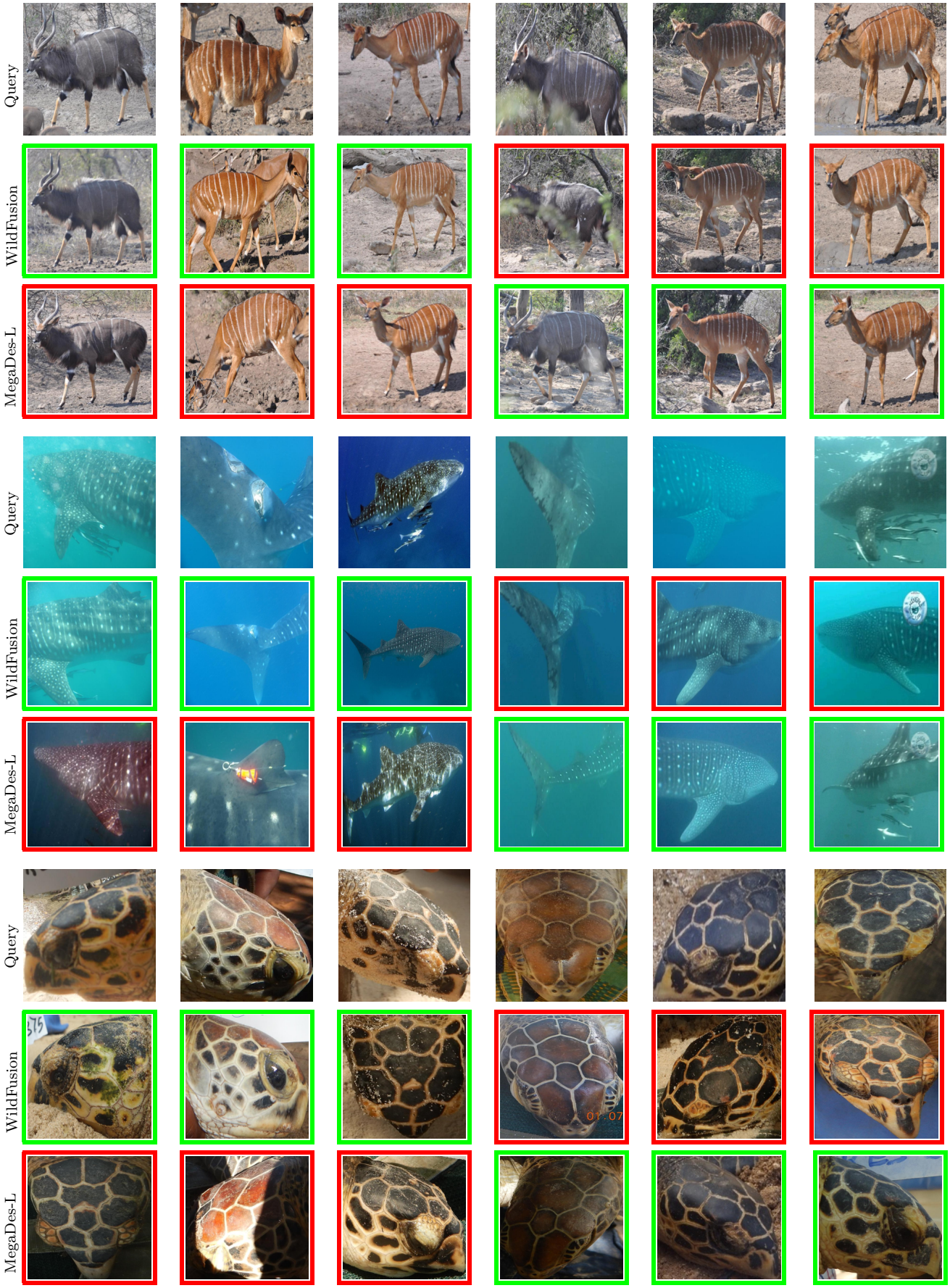


Figure 6.3: **Qualitative performance.** Selected examples where WildFusion changed the decision of the MegaDescriptor-L on NyalaData, WhaleSharkID, and ZindiTurtle; three correct and false samples. We suspect that some wrong matches are mislabeled data.

6.3 Ablation Studies

This section presents a set of ablation studies to empirically validate the design choices behind the WildFusion.

6.3.1 Effect of local matching score threshold

The hyperparameter μ controls the trade-off between low-quality matches and fewer high-quality matches. When μ is low, the score is influenced by many low-quality matches, often presented in the background. When μ is very high, it filters out most of the matches, leading to a loss of information and resulting in a zero score for nearly all pairs.

Comparing performance scores with constant μ and μ selected based on the validation set suggests that local methods are robust to μ selection, and selecting any μ values between [0.4, 0.6] is a good choice. Interestingly, local methods perform better with μ=0.45 fixed for all datasets than searching for optimal μ on the validation set. When local matching scores are combined with global scores, the range of suitable μ values is wider and extends from 0.4 to 0.8. This shows that adding global scores to the ensemble reduces the downside of having zeros in the score for large μ values (see Figure 6.4).

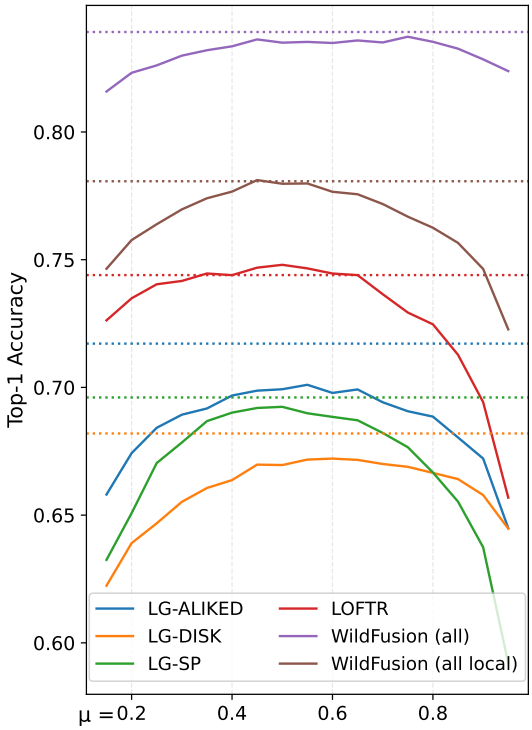


Figure 6.4: **Effect of μ on performance.** Full lines represents constant μ, and dotted lines optimal μ found on validations set for each dataset. Fixing $\mu = 0.5$ provides comparable results to the best μ based on validation set.

6.3.2 Effect of score selection

WildFusion's versatility allows it to fuse any score. As mentioned, using WildFusion only with all local matching scores outperforms the MegaDescriptor-L global score. When we included the global score from the general-purpose feature extractor DinoV2, performance improved only marginally, highlighting the importance of fine-tuning the deep embedding model.

Using the MegaDescriptor-L global score with at least one local matching score significantly outperforms using MegaDescriptor-L alone. Combining it with LG-ALIKED achieves the highest accuracy of 83.0%, followed by LoFTR at 81.4%. LG-SuperPoint and LG-DISK also show comparable performance with accuracies of 80.6% and 81.1%, respectively. Combining the global score with all local matching scores further improves performance, suggesting that the local matching scores are mostly uncorrelated and perform well in the ensemble. More details can be found in Table 6.4.

Table 6.4: **Ablation on local and global score fusion**. We report WildFusion's performance using various local and global methods. Combining local methods with fine-tuned global scores of MegaDetector-L achieves the best results.

Global \ Local	*None*	LG-DISK	LG-SP	LG-ALIKED	LoFTR	all
None	-	68.9	70.1	72.2	74.8	78.5
DINOv2	47.5	70.4	71.6	73.7	74.8	78.8
MegaDescriptor-L	75.5	81.1	80.6	83.0	81.4	84.0

6.3.3 Effect of calibration

Using isotonic regression for calibration yields marginally better results on average compared to logistic regression (83.9% accuracy). However, there is a discrepancy in performance between the datasets. For example, using logistic regression was better on NDD20 (+3.0%) and ZindiTurtleRecall (+ 2.7%), but it significantly underperformed on NyalaData (-6.5%) compared to the isotonic regression. This suggests that the poor performance of WildFusion on ZindiTurtleRecall can be related to incorrect calibration.

How much data do we need for calibration? To test this, we create variously sized subsets of labeled images from the database set and validation set, such that at least two positive and two negative pairs can be created. Pairs created from this subset are used for calibration and finding μ. We perform additional experiments with μ fixed to 0.5 to isolate the effect of low data calibration from finding μ. As visualized in Figure 6.5, isotonic regression performs better than logistic regression in low data scenarios, both with optimized and fixes μ. Calibration with fixed μ is significantly better for a smaller number of samples but yields marginally worse results when a lot of labeled data is available. In general, calibration is very data efficient. For example, fixing μ to 0.5 and calibrating each dataset using only 10 labeled images still gives a reasonable 79.5% accuracy. Adding more data to the calibration further increases the performance of up to 200 samples, where additional data only gives marginal improvements. Our results suggest that WildFusion is viable even with very few labeled samples.

6.3.4 Constraining number of comparisons

Given a database with M samples and a query with N samples, methods based on local features often need to perform pairwise comparisons, needing $M \times N$ comparisons. Since many modern matching algorithms are based on neural networks $M \times N$, neural network inferences are required. With the increasing database size, the computational time quickly becomes unfeasible; therefore, the calculation of all local scores remains a viable option only for moderately sized datasets.

Shortlist strategy We consider a scenario where we have two types of scores, one cheap to calculate, such as global score s_G from MegaDescriptor or DinoV2, and one expensive, such as WildFusion with local matching scores s_W. We follow the shortlist strategy[170] and use cheap global scores to filter candidate samples. The expensive

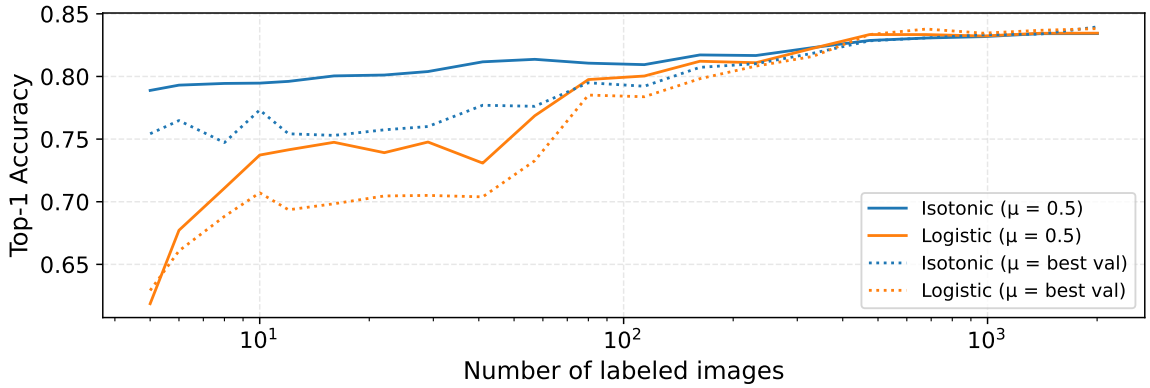


Figure 6.5: **Ablation on optimal number of images for model calibration.** Isotonic regression calibration with fixed $\mu = 0.5$ for all datasets outperforms other approaches in low data scenarios.

scores are calculated for a restricted size shortlist to validate and re-rank the top matches. The running time is controlled by a computational budget B in terms of the number of expensive score evaluations per query image.

Results Using the shortlist strategy, we are efficiently able to utilize the WildFusion scores s_W, which are costly to calculate. On average, budget $B = 300$ is enough to reach accuracy comparable to calculating all scores. For example, on SeaTurtleIdHeads, WildFusion needs only about 200 s_W calculations to reach its peak performance. With a database size of 6063, this results in more than a 30-fold increase in inference speed. Interestingly, performance at $B = 10^3$ is slightly better than using all comparisons. It indicates that local matching scores in WildFusion are prone to some degree of false positive matches when applied to all images in the database. A more detailed visualization of the speed-up is in Figure 6.6.

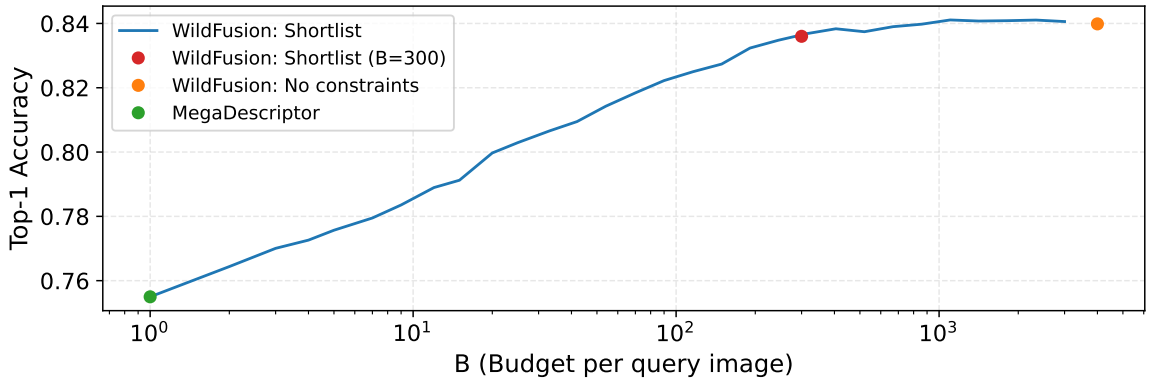


Figure 6.6: **Rate of performance improvement with increasing budget**. The shortlist strategy allows adding more computational resources to improve performance, up to a budget of $B = 200$.

6.4 Zero shot performance

Encouraged by the fact that calibration works well even with a low number of data points to achieve reasonable performance, we conducted an experiment in a zero-shot setting, meaning no data is needed prior to inference. We split datasets into disjoint subsets. For each score, we trained a single calibration model on one subset and evaluated it on a different subset, with $\mu = 0.5$ fixed for all local matching scores. This differs from the default setting, where subsets of the same dataset were used for calibration and evaluation.

For the zero-shot experiment, we evaluated WildFusion using only local matching scores without incorporating MegaDescriptor's global scores, as the latter had already been trained on the data. Our zero-shot WildFusion approach achieved an average accuracy of 76.2%, which is 2.3 percentage points lower than the accuracy obtained with dataset-specific calibration. Notably, this performance is also 0.7 percentage points higher than the state-of-the-art fine-tuned model MegaDescriptor-L-384, demonstrating the effectiveness of our method in a zero-shot setting without fine-tuning or dataset-specific calibration. For more detail about performance, see Table 6.5.

For both novel datasets not included in the MegaDescriptor training set, WildFusion with local scores achieved a perfect 100% accuracy. In contrast, for the CowsDataset, DINOv2 reached an accuracy of 96.0%, while MegaDescriptor achieved 98.7%. Similarly, for the SeaStarReID2023 dataset, DINOv2 obtained an accuracy of 82.2%, whereas MegaDescriptor reached 88.8%.

Table 6.5: **WildFusion performance in zero-shot setting**. No data from the evaluated dataset was used prior to test time (except Wahltinez et al.[160], which used standard classification setting).

			(*local*)	[160]
	MegaDescriptor-L	DINOv2	WildFusion	Wahltinez et al.
CowsDataset	98.7	88.8	100.0	–
SeaStarReID2023	82.2	96.0	100.0	99.9
17 datasets	–	47.5	76.2	–

6.5 Limitations

WildFusion leverages off-the-shelf local matching methods like LoFTR and LightGlue, which were originally trained on datasets featuring static objects. Therefore, it is not optimized for matching animals, where the same animal can be observed in various poses, lighting conditions, and occlusions. Therefore, our work can be extended by adapting or retraining these local feature-matching models specifically for animal identification tasks, potentially improving the accuracy and robustness of the WildFusion approach. The same applies to deep descriptors such as MegaDescriptor and DINOv2, which, if not trained on that species, will most likely underperform.

WildFusion is best suited for offline analysis of existing databases. While we introduced a method to address scalability, it remains insufficient for real-time identification.

Future research could explore the development of more efficient algorithms to enable real-time processing and online identification.

Chapter 7

Conclusion

This thesis aims to improve the automated animal identification field by addressing its challenges and introducing new methods.

We introduce robust time-aware evaluation frameworks alongside new realistic dataset to bridge the gap between laboratory results and real-world performance. Additionally, we emphasized the role of domain-specific knowledge, showing that leveraging anatomical features, such as turtle heads in sea turtle identification, improves model performance.

We propose the **MegaDescriptor** model, a powerful foundation model designed for cross-species animal identification. MegaDescriptor has proven to be effective even on small datasets, outperforming general vision models while being easy to integrate into practical applications. Furthermore, we developed **WildFusion**, a method for animal identification that combines deep metric learning with local feature matching. WildFusion achieves state-of-the-art results while remaining flexible and applicable in zero-shot settings.

To facilitate further research and practical application, we provided essential resources such as the **WildlifeDatasets** library and the **WildlifeTools** library, which we hope will contribute to transparency and reproducibility within the animal identification research community.

7.1 Guidelines for an End-to-End Animal Identification System

Based on our findings, we propose the following guidelines for developing an effective end-to-end automated animal identification system:

1. Use Time-Aware Splits We suggest avoiding splitting the dataset into training and test sets using time-unaware random split as it leads to data leakage and inflated accuracy metrics. We suggest using time-aware evaluation protocols that reflect real-world conditions.

2. Incorporate Domain-Specific Knowledge Using domain-specific knowledge, such as images of relevant anatomical features, can improve identification performance. For example, using segmented turtle heads instead of full bodies improves the identification of turtles.

3. Utilize and Fine-Tune MegaDescriptor Use the pre-trained MegaDescriptor model for robust identification across multiple species. Fine-tune MegaDescriptor when additional labeled data are available to further improve performance.

4. Augment Predictions with Local Feature Matching If slower inference time is not an issue, use WildFusion to combine deep embeddings with local feature-matching techniques. For example, combining MegaDescriptor with even a single local feature matching method such as LoFTR significantly improves identification performance.

5. Aggregate Identifications into Encounters Group images captured during the same encounter to improve identification performance. For example, simple aggregation based on majority voting improves identification performance.

By following these guidelines, an end-to-end animal identification system can be designed to maximize accuracy, generalizability, and real-world usability.

References

[1] Lukas Adam et al. "Exploiting facial side similarities to improve AI-driven sea turtle photo-identification systems". In: *bioRxiv* (2024), pp. 2024–09.

[2] Lukáš Adam et al. "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.

[3] Lukáš Adam et al. "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.

[4] William L Allen and James P Higham. "Assessing the potential information content of multicomponent visual signals: a machine learning approach". In: *Proceedings of the Royal Society B: Biological Sciences* 282.1802 (2015), p. 20142284. URL: https://royalsocietypublishing.org/doi/full/10.1098/rspb.2014.2284.

[5] Carlos JR Anderson et al. "Computer-aided photo-identification system with an application to polar bears based on whisker spot patterns". In: *Journal of Mammalogy* 91.6 (2010), pp. 1350–1359. URL: https://academic.oup.com/jmammal/article/91/6/1350/888329.

[6] Sara Andreotti et al. "An integrated mark-recapture and genetic approach to estimate the population size of white sharks in South Africa". In: *Marine Ecology Progress Series* 552 (2016), pp. 241–253. URL: https://www.int-res.com/abstracts/meps/v552/p241-253/.

[7] William Andrew, Colin Greatwood, and Tilo 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. URL: https://openaccess.thecvf.com/content_ICCV_2017_workshops/w41/html/Andrew_Visual_Localisation_and_ICCV_2017_paper.html.

[8] William Andrew et al. "Automatic individual holstein friesian cattle identification via selective local coat pattern matching in RGB-D imagery". In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2016, pp. 484–488. URL: https://ieeexplore.ieee.org/abstract/document/7532404.

[9] William Andrew et al. "Visual identification of individual Holstein-Friesian cattle via deep metric learning". In: *Computers and Electronics in Agriculture* 185 (2021), p. 106133. URL: https://www.sciencedirect.com/science/article/pii/S0168169921001514.

[10] Gwangbin Bae et al. "DigiFace-1M: 1 Million Digital Face Images for Face Recognition". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 3526–3535.

[11] Cecilia Bardier et al. "Performance of visual vs. software-assisted photo-identification in mark-recapture studies: a case study examining different life stages of the Pacific Horned Frog (Ceratophrys stolzmanni)". In: *Amphibia-Reptilia* 42.1 (2020), pp. 17–28. URL: https://brill.com/view/journals/amre/42/1/article-p17_2.xml.

[12] Anka Bedetti et al. "System for elephant ear-pattern knowledge (SEEK) to identify individual African elephants". In: *Pachyderm* 61 (2020), pp. 63–77. URL: https://pachydermjournal.org/index.php/pachyderm/article/view/65.

[13] *Beluga ID 2022*. 2022. URL: https://lila.science/datasets/beluga-id-2022.

[14] Luca 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. URL: https://ieeexplore.ieee.org/abstract/document/8705934/.

[15] Douglas T Bolger et al. "A computer-assisted system for photographic mark–recapture analysis". In: *Methods in Ecology and Evolution* 3.5 (2012), pp. 813–822.

[16] Alex Borowicz et al. "Social Sensors for Wildlife: Ecological Opportunities in the Era of Camera Ubiquity". In: *Frontiers in Marine Science* (2021), p. 385. URL: https://www.frontiersin.org/articles/10.3389/fmars.2021.645288/full.

[17] Soren Bouma et al. "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. URL: https://ieeexplore.ieee.org/abstract/document/8634778/.

[18] Kay Sara Bradfield. *Photographic identification of individual Archey's frogs, Leiopelma archeyi, from natural markings*. Vol. 191. Department of Conservation Wellington, New Zealand, 2004. URL: https://www.doc.govt.nz/globalassets/documents/science-and-technical/dsis191.pdf.

[19] Daniel Brenot. *I3S*. 2018. URL: https://github.com/daniel-brenot/I3S-Interactive-Individual-Identification-System-Desktop.

[20] Jane Bromley et al. "Signature verification using a" siamese" time delay neural network". In: *Advances in neural information processing systems* 6 (1993).

[21] Otto Brookes and Tilo Burghardt. "A dataset and application for facial recognition of individual gorillas in zoo environments". In: *arXiv preprint arXiv:2012.04689* (2020). URL: https://arxiv.org/abs/2012.04689.

[22] Joakim Bruslund Haurum et al. "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. URL: https://ieeexplore.ieee.org/document/9096922.

[23] Clemens-Alexander Brust et al. "Towards automated visual monitoring of individual gorillas in the wild". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 2820–2830. URL: https://openaccess.thecvf.com/content_ICCV_2017_workshops/w41/html/Brust_Towards_Automated_Visual_ICCV_2017_paper.html.

[24] Tilo Burghardt et al. "Automated visual recognition of individual african penguins". In: *Fifth International Penguin Conference*. University of Surrey. 2004. URL: https://openresearch.surrey.ac.uk/esploro/outputs/conferencePresentation/Automated-visual-recognition-of-individual-African-penguins/99516642902346.

[25] Giovanni Caci et al. "Spotting the right spot: computer-aided individual identification of the threatened cerambycid beetle Rosalia alpina". In: *Journal of insect conservation* 17.4 (2013), pp. 787–795. URL: https://link.springer.com/article/10.1007/s10841-013-9561-0.

[26] Alice S Carpentier et al. "Stability of facial scale patterns on green sea turtles Chelonia mydas over time: a validation for the use of a photo-identification method". In: *Journal of Experimental Marine Biology and Ecology* 476 (2016), pp. 15–21. URL: https://www.sciencedirect.com/science/article/pii/S0022098115300733.

[27] Alice S Carpentier et al. "Stability of facial scale patterns on green sea turtles Chelonia mydas over time: A validation for the use of a photo-identification method". In: *Journal of Experimental Marine Biology and Ecology* 476 (2016), pp. 15–21.

[28] Alecia J Carter et al. "Structured association patterns and their energetic benefits in female eastern grey kangaroos, Macropus giganteus". In: *Animal Behaviour* 77.4 (2009), pp. 839–846. URL: https://www.sciencedirect.com/science/article/pii/S0003347208005708.

[29] Steven JB Carter et al. "Automated marine turtle photograph identification using artificial neural networks, with application to green turtles". In: *Journal of experimental marine biology and ecology* 452 (2014), pp. 105–110.

[30] *Cat Individual Images*. 2019. URL: https://www.kaggle.com/datasets/timost1234/cat-individuals.

[31] Vojtěch Čermák et al. "WildlifeDatasets: An open-source toolkit for animal re-identification". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 5953–5963.

[32] Aditya Chattopadhay et al. "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks". In: *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2018, pp. 839–847.

[33] Gullal Singh Cheema and Saket Anand. "Automatic detection and recognition of individuals in patterned species". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2017, pp. 27–38. URL: https://link.springer.com/chapter/10.1007/978-3-319-71273-4_3.

[34] Ted Cheeseman et al. "Advanced image recognition: a fully automated, high-accuracy photo-identification matching system for humpback whales". In: *Mammalian Biology* 102.3 (2022), pp. 915–929. URL: https://doi.org/10.1007/s42991-021-00180-9.

[35] Kai Chen et al. "Hybrid task cascade for instance segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4974–4983.

[36] Kai Chen et al. "MMDetection: Open MMLab Detection Toolbox and Benchmark". In: *arXiv preprint arXiv:1906.07155* (2019).

[37] Peng Chen et al. "A study on giant panda recognition based on images of a large proportion of captive pandas". In: *Ecology and Evolution* 10.7 (2020), pp. 3561–3573. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.6152.

[38] Weihua Chen et al. "Beyond triplet loss: a deep quadruplet network for person re-identification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 403–412.

[39] Bowen Cheng, Alex Schwing, and Alexander Kirillov. "Per-pixel classification is not all you need for semantic segmentation". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 17864–17875.

[40] Sumit Chopra, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 539–546.

[41] Melanie Clapham et al. "Automated facial recognition for wildlife that lack unique markings: A deep learning approach for brown bears". In: *Ecology and evolution* 10.23 (2020), pp. 12883–12892.

[42] Irina Clavadetscher et al. "Development of an image-based body condition score for giraffes Giraffa camelopardalis and a comparison of zoo-housed and free-ranging individuals". In: *Journal of Zoo and Aquarium Research* 9.3 (2021), pp. 170–185. URL: https://w.jzar.org/jzar/article/view/615.

[43] Corinna Cortes. "Support-Vector Networks". In: *Machine Learning* (1995).

[44] Jonathan P Crall et al. "Hotspotter—patterned species instance recognition". In: *2013 IEEE workshop on applications of computer vision (WACV)*. IEEE. 2013, pp. 230–237.

[45] Debayan 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. URL: https://ieeexplore.ieee.org/abstract/document/8698538/.

[46] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 248–255.

[47] Jiankang Deng et al. "Arcface: Additive angular margin loss for deep face recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4690–4699.

[48] Jiankang Deng et al. "Sub-center arcface: Boosting face recognition by large-scale noisy web faces". In: *European Conference on Computer Vision*. Springer. 2020, pp. 741–757.

[49] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "Superpoint: Self-supervised interest point detection and description". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 224–236.

[50] Nkosikhona Dlamini and Terence L van Zyl. "Automated Identification of Individuals in Wildlife Population Using Siamese Neural Networks". In: *2020 7th International Conference on Soft Computing & Machine Intelligence (ISCMI)*. IEEE. 2020, pp. 224–228.

[51] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[52] Axel Drechsler, Tobias Helling, and Sebastian Steinfartz. "Genetic fingerprinting proves cross-correlated automatic photo-identification of individuals as highly efficient in large capture–mark–recapture studies". In: *Ecology and Evolution* 5.1 (2015), pp. 141–151. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ece3.1340.

[53] Stephen G Dunbar et al. "HotSpotter: Using a computer-driven photo-id application to identify sea turtles". In: *Journal of Experimental Marine Biology and Ecology* 535 (2021), p. 151490. URL: https://www.sciencedirect.com/science/article/pii/S0022098120301738.

[54] James Duyck et al. "Sloop: A pattern retrieval engine for individual animal identification". In: *Pattern Recognition* 48.4 (2015), pp. 1059–1073. URL: https://www.sciencedirect.com/science/article/pii/S0031320314002763.

[55] Charlotte Faul, Norman Wagner, and Michael Veith. "Successful automated photographic identification of larvae of the European Fire Salamander, Salamandra salamandra". In: *SALAMANDRA* 58.1 (2022), pp. 52–63. URL: https://www.salamandra-journal.com/index.php/home/contents/2071-faul-c-n-wagner-m-veith.

[56] André C Ferreira et al. "Deep learning-based methods for individual recognition in small birds". In: *Methods in Ecology and Evolution* 11.9 (2020), pp. 1072–1085.

[57] Alexander Freytag et al. "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.

[58] Ashley J Frisch and Jean-Paul A Hobbs. "Photographic identification based on unique, polymorphic colour patterns: a novel method for tracking a marine crustacean". In: *Journal of Experimental Marine Biology and Ecology* 351.1-2 (2007), pp. 294–299. URL: https://www.sciencedirect.com/science/article/pii/S0022098107003401.

[59] Frederick N Fritsch and Ralph E Carlson. "Monotone piecewise cubic interpolation". In: *SIAM Journal on Numerical Analysis* 17.2 (1980), pp. 238–246.

[60] Lili Fu and Gong He. *Cow dataset*. 2021. URL: https://doi.org/10.6084/m9.figshare.16879780.v1.

[61] Jing Gao et al. "Towards Self-Supervision for Video Identification of Individual Holstein-Friesian Cattle: The Cows2021 Dataset". In: *arXiv preprint arXiv:2105.01938* (2021). URL: https://arxiv.org/abs/2105.01938.

[62] Riana Zanarivero Gardiner et al. "A face in the crowd: a non-invasive and cost effective photo-identification methodology to understand the fine scale movement of eastern water dragons". In: *PloS one* 9.5 (2014), e96992. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0096992.

[63] Tilen Genov et al. "Novel method for identifying individual cetaceans using facial features and symmetry: A test case using dolphins". In: *Marine Mammal Science* 34.2 (2018), pp. 514–528. URL: https://onlinelibrary.wiley.com/doi/10.1111/mms.12451.

[64] Andrew Gilman et al. "Computer-assisted recognition of dolphin individuals using dorsal fin pigmentations". In: *2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE. 2016, pp. 1–6. URL: https://ieeexplore.ieee.org/abstract/document/7804460.

[65] Raia Hadsell, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant mapping". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE. 2006, pp. 1735–1742.

[66] Mark F Hansen et al. "Towards on-farm pig face recognition using convolutional neural networks". In: *Computers in Industry* 98 (2018), pp. 145–152. URL: https://www.sciencedirect.com/science/article/pii/S0166361517304992.

[67] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[68] Kaiming He et al. "Mask R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017.

[69] Qi He et al. "Distinguishing individual red pandas from their faces". In: *Chinese conference on pattern recognition and computer vision (PRCV)*. Springer. 2019, pp. 714–724. URL: https://link.springer.com/chapter/10.1007/978-3-030-31723-2_61.

[70] Zhimin He et al. "Animal Re-Identification Algorithm for Posture Diversity". In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, pp. 1–5.

[71] Alexander Hermans, Lucas Beyer, and Bastian Leibe. "In defense of the triplet loss for person re-identification". In: *arXiv preprint arXiv:1703.07737* (2017).

[72] Jason Holmberg, Bradley Norman, and Zaven Arzoumanian. "Estimating population size, structure, and residency time for whale sharks Rhincodon typus through collaborative photo-identification". In: *Endangered Species Research* 7.1 (2009), pp. 39–53. URL: https://www.int-res.com/abstracts/esr/v7/n1/p39-53/.

[73] Gary B Huang and Erik Learned-Miller. *Labeled Faces in the Wild: Updates and New Reporting Procedures*. Tech. rep. UM-CS-2014-003. University of Massachusetts, Amherst, 2014.

[74] Gary B Huang et al. "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments". In: *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*. 2008. URL: https://hal.inria.fr/inria-00321923/.

[75] Christine L Huffard et al. "Individually unique body color patterns in octopus (Wunderpus photogenicus) allow for photoidentification". In: *PloS one* 3.11 (2008), e3732. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0003732.

[76] Benjamin Hughes and Tilo Burghardt. "Automated visual fin identification of individual great white sharks". In: *International Journal of Computer Vision* 122.3 (2017), pp. 542–557. URL: https://link.springer.com/article/10.1007/s11263-016-0961-y.

[77] *Humpback Whale Identification*. 2019. URL: https://www.kaggle.com/competitions/humpback-whale-identification.

[78] Emmanuel Kabuga. "Using neural networks to identify individual animals from photographs". MA thesis. Faculty of Science, 2019.

[79] Marcella J Kelly. "Computer-aided photograph matching in studies using individual identification: an example from Serengeti cheetahs". In: *Journal of Mammalogy* 82.2 (2001), pp. 440–449.

[80] Ira Kemelmacher-Shlizerman et al. "The megaface benchmark: 1 million faces for recognition at scale". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4873–4882. URL: https://openaccess.thecvf.com/content_cvpr_2016/html/Kemelmacher-Shlizerman_The_MegaFace_Benchmark_CVPR_2016_paper.html.

[81] Daria Kern et al. "Towards Automated Chicken Monitoring: Dataset and Machine Learning Methods for Visual, Noninvasive Reidentification". In: *Animals* 15.1 (2025). ISSN: 2076-2615. DOI: 10.3390/ani15010001. URL: https://www.mdpi.com/2076-2615/15/1/1.

[82] Alexander Kirillov et al. "Segment anything". In: *arXiv preprint arXiv:2304.02643* (2023).

[83] Carey D Knox, Alison Cree, and Philip J Seddon. "Accurate identification of individual geckos (Naultinus gemmeus) through dorsal pattern differentiation". In: *New Zealand Journal of Ecology* (2013), pp. 60–66. URL: https://www.jstor.org/stable/24060758.

[84] Dmitry A Konovalov et al. "Individual minke whale recognition using deep learning convolutional neural networks". In: *Journal of Geoscience and Environment Protection* 6 (2018), pp. 25–36. URL: https://researchonline.jcu.edu.au/54297/.

[85] Matthias Korschens and Joachim Denzler. "ELPephants: A fine-grained dataset for elephant re-identification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 263–270.

[86] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[87] Thomas E Kucera. "Social behavior and breeding system of the desert mule deer". In: *Journal of Mammalogy* 59.3 (1978), pp. 463–476. URL: https://academic.oup.com/jmammal/article-abstract/59/3/463/852096.

[88] Peter Kulits et al. "ElephantBook: A Semi-Automated Human-in-the-Loop System for Elephant Re-Identification". In: *ACM SIGCAS Conference on Computing and Sustainable Societies*. 2021, pp. 88–98. URL: https://dl.acm.org/doi/abs/10.1145/3460112.3471947.

[89] Mayank Lahiri et al. "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.

[90] Izzy Langley, Emily Hague, and Mònica Arso Civil. "Assessing the performance of open-source, semi-automated pattern recognition software for harbour seal (P. v. vitulina) photo ID". In: *Mammalian Biology* (2021), pp. 1–10. URL: https://link.springer.com/article/10.1007/s42991-021-00165-8.

[91] Catherine A Langtimm et al. "Survival estimates for Florida manatees from the photo-identification of individuals". In: *Marine Mammal Science* 20.3 (2004), pp. 438–463. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1748-7692.2004.tb01171.x.

[92] Yann LeCun, Corinna Cortes, and CJ Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010).

[93] Shijun Li et al. "Individual dairy cow identification based on lightweight convolutional neural network". In: *Plos one* 16.11 (2021), e0260510.

[94] Shuyuan Li et al. "ATRW: A Benchmark for Amur Tiger Re-identification in the Wild". In: *Proceedings of the 28th ACM International Conference on Multimedia*. Association for Computing Machinery, 2020, 2590–2598.

[95] Zhengqi Li and Noah Snavely. "Megadepth: Learning single-view depth prediction from internet photos". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2041–2050.

[96] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. "LightGlue: Local Feature Matching at Light Speed". In: *arXiv preprint arXiv:2306.13643* (2023).

[97] Weiyang Liu et al. "Sphereface: Deep hypersphere embedding for face recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 212–220.

[98] Ze Liu et al. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.

[99] Zhuang Liu et al. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11976–11986.

[100] Alexander Loos and Andreas Ernst. "An automated chimpanzee identification system using face detection and recognition". In: *EURASIP Journal on Image and Video Processing* 2013.1 (2013), pp. 1–17. URL: https://link.springer.com/article/10.1186/1687-5281-2013-49.

[101] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60 (2004), pp. 91–110.

[102] Dimitris Margaritoulis and Aliki Panagopoulou. "Greece". In: *Sea turtles in the Mediterranean: distribution, threats and conservation priorities*. Ed. by P. Casale. IUCN, 2010, pp. 85–113.

[103] Dimitris Margaritoulis et al. "Reproductive Longevity of Loggerhead Sea Turtles Nesting in Greece". In: *Chelonian Conservation and Biology* 19.1 (2020), pp. 133–136. URL: https://doi.org/10.2744/CCB-1437.1.

[104] Shaun D McConkey. "Photographic identification of the New Zealand sea lion: A new technique". In: *New Zealand Journal of Marine and Freshwater Research* 33.1 (1999), pp. 63–66. URL: https://www.tandfonline.com/doi/abs/10.1080/00288330.1999.9516857.

[105] Ricardo de Sá Rocha Mello et al. "Comparison among three body parts and three software packages to optimise photographic identification of a reptile (tuatara, Sphenodon punctatus)". In: *Amphibia-Reptilia* 40.2 (2019), pp. 233–244. URL: https://brill.com/view/journals/amre/40/2/article-p233_8.xml.

[106] Onoufrios Mettouris, George Megremis, and Sinos Giokas. "A newt does not change its spots: using pattern mapping for the identification of individuals in large populations of newt species". In: *Ecological Research* 31.3 (2016), pp. 483–489. URL: https://link.springer.com/article/10.1007/s11284-016-1346-y.

[107] Vincent Miele et al. "Revisiting animal photo-identification using deep metric learning and network analysis". In: *Methods in Ecology and Evolution* 12.5 (2021), pp. 863–873.

[108] Sophie K Mills et al. "Photo identification for sea turtles: Flipper scales more accurate than head scales using APHIS". In: *Journal of Experimental Marine Biology and Ecology* 566 (2023), p. 151923.

[109] Thierry Pinheiro Moreira et al. "Where is my puppy? Retrieving lost dogs by facial features". In: *Multimedia Tools and Applications* 76.14 (2017), pp. 15325–15340.

[110] Dorian Moro and Isobel MacAulay. "Computer-aided pattern recognition of large reptiles as a noninvasive application to identify individuals". In: *Journal of Applied Animal Welfare Science* 17.2 (2014), pp. 125–135. URL: https://www.tandfonline.com/doi/full/10.1080/10888705.2014.883925.

[111] Thomas A Morrison et al. "Estimating survival in photographic capture–recapture studies: overcoming misidentification error". In: *Methods in Ecology and Evolution* 2.5 (2011), pp. 454–463. URL: https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/j.2041-210X.2011.00106.x.

[112] Olga Moskvyak et al. "Robust re-identification of manta rays from natural markings by learning pose invariant embeddings". In: *2021 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE. 2021, pp. 1–8. URL: https://ieeexplore.ieee.org/abstract/document/9647359/.

[113] Guillaume Mougeot, Dewei Li, and Shuai Jia. "A Deep Learning Approach for Dog Face Verification and Recognition". In: *PRICAI 2019: Trends in Artificial Intelligence*. Ed. by Abhaya C. Nayak and Alok Sharma. Cham: Springer International Publishing, 2019, pp. 418–430. ISBN: 978-3-030-29894-4.

[114] Oscar Moya et al. "APHIS: a new software for photo-matching in ecological studies". In: *Ecological informatics* 27 (2015), pp. 64–70. URL: https://www.sciencedirect.com/science/article/pii/S1574954115000680.

[115] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. "A metric learning reality check". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*. Springer. 2020, pp. 681–699.

[116] Ekaterina Nepovinnykh et al. "SealID: Saimaa ringed seal re-identification dataset". In: *Sensors* 22.19 (2022), p. 7602.

[117] Hong-Wei Ng and Stefan Winkler. "A data-driven approach to cleaning large face datasets". In: *2014 IEEE international conference on image processing (ICIP)*. IEEE. 2014, pp. 343–347.

[118] Robert B Nipko, Brogan E Holcombe, and Marcella J Kelly. "Identifying individual jaguars and ocelots via pattern-recognition software: comparing HotSpotter and Wild-ID". In: *Wildlife Society Bulletin* 44.2 (2020), pp. 424–433. URL: https://wildlife.onlinelibrary.wiley.com/doi/full/10.1002/wsb.1086.

[119] Maxime Oquab et al. "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193* (2023).

[120] Lasha Otarashvili. *MiewID*. 2023. DOI: 10.5281/zenodo.13647526. URL: https://github.com/WildMeOrg/wbia-plugin-miew-id.

[121] Kostas Papafitsoros, Aliki Panagopoulou, and Gail Schofield. "Social media reveals consistently disproportionate tourism pressure on a threatened marine vertebrate". In: *Animal Conservation* 24.4 (2021), pp. 568–579. URL: https://doi.org/10.1111/acv.12656.

[122] Kostas Papafitsoros et al. "SeaTurtleID: A novel long-span dataset highlighting the importance of timestamps in wildlife re-identification". In: *arXiv preprint arXiv:2211.10307* (2022).

[123] Jason Remington Parham et al. "Animal population censusing at scale with citizen science and photographic identification". In: *2017 AAAI Spring Symposium Series*. 2017. URL: https://www.aaai.org/ocs/index.php/SSS/SSS17/paper/viewPaper/15245.

[124] MDM Pawley et al. "Examining the viability of dorsal fin pigmentation for individual identification of poorly-marked delphinids". In: *Scientific reports* 8.1 (2018), pp. 1–12. URL: https://www.nature.com/articles/s41598-018-30842-7.

[125] Malte Pedersen et al. "Re-Identification of Giant Sunfish using Keypoint Matching". In: *Proceedings of the Northern Lights Deep Learning Workshop*. Vol. 3. 2022.

[126] Daniele Pellitteri-Rosa et al. "Photographic identification in reptiles: a matter of scales". In: *Amphibia-Reptilia* 31.4 (2010), pp. 489–502. URL: https://brill.com/view/journals/amre/31/4/article-p489_6.xml.

[127] John Platt et al. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". In: *Advances in large margin classifiers* 10.3 (1999), pp. 61–74.

[128] Jouke Prop, Arnstein Staverløkk, and Børge Moe. "Identifying individual polar bears at safe distances: A test with captive animals". In: *PloS one* 15.2 (2020), e0228991. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0228991.

[129] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[130] Prashanth C Ravoor and TSB Sudarshan. "Deep Learning Methods for Multi-Species Animal Re-identification and Tracking–a Survey". In: *Computer Science Review* 38 (2020), p. 100289. URL: https://www.sciencedirect.com/science/article/pii/S1574013720303890.

[131] Julien Renet et al. "Monitoring amphibian species with complex chromatophore patterns: a non-invasive approach with an evaluation of software effectiveness and reliability". In: *Herpetological Journal* 29 (2019), pp. 13–22. URL: https://hal.archives-ouvertes.fr/hal-02408090/.

[132] Vito Renò et al. "A SIFT-based software system for the photo-identification of the Risso's dolphin". In: *Ecological informatics* 50 (2019), pp. 95–101. URL: https://www.sciencedirect.com/science/article/pii/S1574954118301377.

[133] *Right Whale Recognition*. 2015. URL: https://www.kaggle.com/c/noaa-right-whale-recognition.

[134] Federico Romiti et al. "Photographic identification method (PIM) using natural body marks: A simple tool to make a long story short". In: *Zoologischer Anzeiger* 266 (2017), pp. 136–147. URL: https://www.sciencedirect.com/science/article/pii/S0044523116301322.

[135] Ethan Rublee et al. "ORB: An efficient alternative to SIFT or SURF". In: *2011 International conference on computer vision*. Ieee. 2011, pp. 2564–2571.

[136] Paul-Edouard Sarlin et al. "Superglue: Learning feature matching with graph neural networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4938–4947.

[137] Jonathan Schneider et al. "Can Drosophila melanogaster tell who's who?" In: *PloS one* 13.10 (2018), e0205043.

[138] Stefan Schneider, Graham W Taylor, and Stefan C Kremer. "Similarity learning networks for animal individual re-identification: an ecological perspective". In: *Mammalian Biology* (2022), pp. 1–16.

[139] Stefan Schneider et al. "Past, present and future approaches using computer vision for animal re-identification from camera trap data". In: *Methods in Ecology and Evolution* 10.4 (2019), pp. 461–470. URL: https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.13133.

[140] Daniel Schofield et al. "Chimpanzee face recognition from videos in the wild using deep learning". In: *Science advances* 5.9 (2019), eaaw0736. URL: https://www.science.org/doi/full/10.1126/sciadv.aaw0736.

[141] Gail Schofield et al. "Investigating the viability of photo-identification as an objective tool to study endangered sea turtle populations". In: *Journal of Experimental Marine Biology and Ecology* 360.2 (2008), pp. 103–108. URL: https://doi.org/10.1016/j.jembe.2008.04.005.

[142] Gail Schofield et al. "Long-term photo-id and satellite tracking reveal sex-biased survival linked to movements in an endangered species". In: *Ecology* 11 (7 2020), e03027. URL: https://doi.org/10.1002/ecy.3027.

[143] Gail Schofield et al. "More aggressive sea turtles win fights over foraging resources independent of body size and years of presence". In: *Animal Behaviour* 190 (2022), pp. 209–219. URL: https://www.sciencedirect.com/science/article/pii/S0003347222001312.

[144] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 815–823.

[145] Risa Shinoda and Kaede Shiohara. "PetFace: A Large-Scale Dataset and Benchmark for Animal Identification". In: *arXiv preprint arXiv:2407.13555* (2024).

[146] Samuel Stevens et al. "Bioclip: A vision foundation model for the tree of life". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 19412–19424.

[147] Jiaming Sun et al. "LoFTR: Detector-free local feature matching with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 8922–8931.

[148] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[149] Frederic Tausch et al. "Bumblebee re-identification dataset". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*. 2020, pp. 35–37. URL: https://ieeexplore.ieee.org/document/9096909.

[150] Daniel H Thornton and Charles E Pekins. "Spatially explicit capture–recapture analysis of bobcat (Lynx rufus) density: implications for mesocarnivore monitoring". In: *Wildlife Research* 42.5 (2015), pp. 394–404. URL: https://www.publish.csiro.au/wr/wr15092.

[151] Christopher Town, Andrea Marshall, and Nutthaporn Sethasathien. "Manta Matcher: automated photographic identification of manta rays using keypoint features". In: *Ecology and evolution* 3.7 (2013), pp. 1902–1914. URL: https://onlinelibrary.wiley.com/doi/10.1002/ece3.587.

[152] Cameron Trotter et al. "NDD20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation". In: *arXiv preprint arXiv:2005.13359* (2020). URL: https://arxiv.org/abs/2005.13359.

[153] Botswana Predator Conservation Trust. *Panthera pardus CSV custom export*. 2022. URL: https://africancarnivore.wildbook.org/.

[154] Matthew Turk and Alex Pentland. "Eigenfaces for recognition". In: *Journal of cognitive neuroscience* 3.1 (1991), pp. 71–86.

[155] *Turtle Recall: Conservation Challenge*. 2022. URL: https://zindi.africa/competitions/turtle-recall-conservation-challenge.

[156] Reny Blue Tyson Moore et al. "Rise of the Machines: Best Practices and Experimental Evaluation of Computer-Assisted Dorsal Fin Image Matching Systems for Bottlenose Dolphins". In: *Frontiers in Marine Science* (2022), p. 445. URL: https://www.frontiersin.org/articles/10.3389/fmars.2022.849813/full.

[157] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. "DISK: Learning local features with policy gradient". In: *Advances in Neural Information Processing Systems* 33 (2020).

[158] Masataka Ueno et al. "Automatic individual recognition of Japanese macaques (Macaca fuscata) from sequential images". In: *Ethology* 128.5 (2022), pp. 461–470. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/eth.13277.

[159] Maxime Vidal et al. "Perspectives on individual animal identification from biology and computer vision". In: *Integrative and comparative biology* 61.3 (2021), pp. 900–916.

[160] Oscar Wahltinez and Sarah J Wahltinez. "An open-source general purpose machine learning framework for individual animal re-identification using few-shot learning". In: *Methods in Ecology and Evolution* 15.2 (2024), pp. 373–387.

[161] Le Wang et al. "Giant panda identification". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 2837–2849. URL: https://ieeexplore.ieee.org/document/9347819.

[162] Le Wang et al. "Giant Panda Identification". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 2837–2849. DOI: 10.1109/TIP.2021.3055627.

[163] Hendrik Weideman et al. "Extracting identifying contours for African elephants and humpback whales using a learned appearance model". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 1276–1285.

[164] Hendrik 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.

[165] Ross Wightman. *PyTorch Image Models*. https://github.com/rwightman/pytorch-image-models. 2019. DOI: 10.5281/zenodo.4414861.

[166] Claire L Witham. "Automated face recognition of rhesus macaques". In: *Journal of Neuroscience Methods* 300 (2018), pp. 157–165. URL: https://www.sciencedirect.com/science/article/pii/S0165027017302637.

[167] Saining Xie et al. "Aggregated residual transformations for deep neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, 2017, pp. 1492–1500.

[168] Cheng Yan et al. "Beyond triplet loss: person re-identification with fine-grained difference-aware pairwise loss". In: *IEEE Transactions on Multimedia* 24 (2021), pp. 1665–1677.

[169] Daode Yang et al. "Using head patch pattern as a reliable biometric character for noninvasive individual recognition of an endangered pitviper Protobothrops mangshanensis". In: *Asian Herpetological Research* 4.2 (2013), pp. 134–139. URL: http://www.ahr-journal.com/en/oa/DArticle.aspx?type=view&id=20130007.

[170] Hantao Yao et al. "Large-scale person re-identification as retrieval". In: *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2017, pp. 1440–1445.

[171] Guoshen Yu and Jean-Michel Morel. "ASIFT: An algorithm for fully affine invariant comparison". In: *Image Processing On Line* 1 (2011), pp. 11–38.

[172] Bianca Zadrozny and Charles Elkan. "Transforming classifier scores into accurate multiclass probability estimates". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 694–699.

[173] Tingting Zhang et al. "YakReID-103: A Benchmark for Yak Re-Identification". In: *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2021, pp. 1–8. URL: https://ieeexplore.ieee.org/abstract/document/9484341/.

[174] Xiaoming Zhao et al. "ALIKED: A Lighter Keypoint and Descriptor Extraction Network via Deformable Transformation". In: *IEEE Transactions on Instrumentation & Measurement* 72 (2023), pp. 1–16. DOI: 10.1109/TIM.2023.3271000. URL: https://arxiv.org/pdf/2304.03608.pdf.

[175] Thi Thi Zin et al. "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. URL: http://www.iaeng.org/publication/IMECS2018/IMECS2018_pp320-323.pdf.

[176] Matthias Zuerl et al. "PolarBearVidID: A Video-Based Re-Identification Benchmark Dataset for Polar Bears". In: *Animals* 13.5 (2023), p. 801.

[177] Silvia Zuffi, Angjoo Kanazawa, and Michael J Black. "Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 3955–3963. URL: https://openaccess.thecvf.com/content_cvpr_2018/html/Zuffi_Lions_and_Tigers_CVPR_2018_paper.html.

[178] Silvia Zuffi et al. "Three-D Safari: Learning to Estimate Zebra Pose, Shape, and Texture from Images "In the Wild"". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5359–5368. URL: https://ieeexplore.ieee.org/document/9010937.