WildlifeDatasets: An open-source toolkit for animal re-identification

Vojtěch Čermák[1], Lukas Picek[2,3], Lukáš Adam[1] and Kostas Papafitsoros[4]
[1]Czech Technical University in Prague, [2]University of West Bohemia, [3]INRIA
[4]Queen Mary University of London
cermavo3@fel.cvut.cz, picekl@kky.zcu.cz/lpicek@inria.cz
lukas.adam.cr@gmail.com, k.papafitsoros@qmul.ac.uk

Abstract

In this paper, we present WildlifeDatasets – an open-source toolkit intended primarily for ecologists and computer-vision / machine-learning researchers. The WildlifeDatasets is written in Python, allows straightforward access to publicly available wildlife datasets, and provides a wide variety of methods for dataset pre-processing, performance analysis, and model fine-tuning. We showcase the toolkit in various scenarios and baseline experiments, including, to the best of our knowledge, the most comprehensive experimental comparison of datasets and methods for wildlife re-identification, including both local descriptors and deep learning approaches. Furthermore, we provide the first-ever foundation model for individual re-identification within a wide range of species – MegaDescriptor – that provides state-of-the-art performance on animal re-identification datasets and outperforms other pretrained models such as CLIP and DINOv2 by a significant margin. To make the model available to the general public and to allow easy integration with any existing wildlife monitoring applications, we provide multiple MegaDescriptor flavors (i.e., Small, Medium, and Large) through the HuggingFace hub.

1. Introduction

Animal re-identification is essential for studying different aspects of wildlife, like population monitoring, movements, behavioral studies, and wildlife management [39,45,50]. While the precise definition and approaches to animal re-identification may vary in the literature, the objective remains consistent. The main goal is to accurately and efficiently recognize individual animals within one species based on their unique characteristics, e.g., markings, patterns, or other distinctive features.

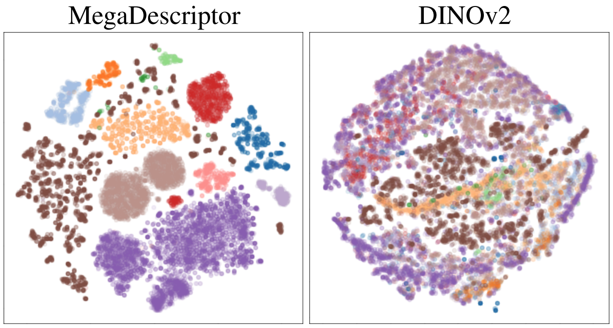


Figure 1. **Latent space separability of MegaDescriptor**. Embedding visualization (t-sne) of unseen individual animals (identity-wise) for the proposed MegaDescriptor and DINOv2. Colors represent different datasets (i.e., species).

Automatizing the identification and tracking of individual animals enables the collection of precise and extensive data on population dynamics, migration patterns, habitat usage, and behavior, facilitating researchers in monitoring movements, evaluating population sizes, and observing demographic shifts. This invaluable information contributes to a deeper comprehension of species dynamics, identifying biodiversity threats, and developing conservation strategies grounded in evidence.

Similarly, the increasing sizes of the collected data and the increasing demand for manual (i.e., time-consuming) processing of the data highlighted the need for automated methods to reduce labor-intensive human supervision in individual animal identification. As a result, a large number of automatic re-identification datasets and methods have been developed, covering several animal groups like primates [23, 54], carnivores [18, 31, 48], reptiles [4, 21], whales [1, 2, 13], and mammals [3, 47, 57].

However, there is a lack of standardization in algorithmic procedures, evaluation metrics, and dataset utilization across the literature. This hampers the comparability and reproducibility of results, hindering the progress of the field. It is, therefore, essential to categorize and re-evaluate general re-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. This analysis will aid researchers and practitioners in selecting suitable algorithms for their specific re-identification needs, ultimately advancing the field of animal re-identification and its applications in wildlife conservation and research.

To address these issues, we have developed an open-source toolkit – WildlifeDatasets – intended primarily for ecologists and computer-vision / machine-learning researchers. In this paper, besides the short description of the main features of our tool, (i) we list all publicly available wildlife re-identification datasets, (ii) perform the largest experimental comparison of datasets and wildlife re-identification methods, (iii) describe a foundation model – MegaDescriptor – based on different Swin architectures and trained on a newly comprised dataset, and (iv) provide a variety of pre-trained models on a HuggingFace hub.

2. Related work

Similarly, as in other fields, the development of methods and datasets for automated animal re-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 re-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 [6, 10, 20, 25, 31, 52], 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.

2.1. Tools and methods

There are three primary approaches commonly used for wildlife re-identification – (i) local descriptors [9, 21, 43], (ii) deep descriptors [12, 16, 31, 34, 49], and (iii) species-specific methods [6, 10, 25, 29, 52].

Local-feature-based 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. 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 [42] or DINOv2 [37], etc.

Even though approaches based on SIFT, SURF, or ORB descriptors exhibit limitations in scaling efficiently to larger datasets and their performance, all available software products, e.g., WildID [11], HotSpotter [15], and I^{3}S, 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).

Deep feature-based approaches are based on vector representation of the image learned through optimizing a deep neural network. Similarly, as in local feature-based methods, the resulting deep embedding vector (usually 1024 or 2048d) is matched with an identity database.

Applying deep learning to wildlife re-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 (e.g., CLIP [42] and DINOv2 [37]). These models are primarily designed for general computer vision tasks. Therefore, they are not adapted nor tested for the nuances of wildlife re-identification, which heavily relies on fine-grained features.

Species-specific methods are tailored to an individual species or groups of closely related species, particularly those with visually distinct patterns. These methods typically focus on visual characteristics unique to the target species, restricting their applicability beyond the species they were developed for. Moreover, they often entail substantial manual preprocessing steps, such as extracting patches from regions of interest or accurately aligning compared images. For instance, one such approach involves employing Chamfer distance to measure the distance between greyscale patterns in polar bear whiskers [6]. Other examples include computing correlation between aligned patches derived from cheetah spots [29] or similarity between two images based on the count of matching pixels within newt patterns [20].

3. The WildlifeDatasets toolkit

One of the current challenges for the advancement of wildlife re-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 re-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.

3.1. All publicly available wildlife datasets at hand

The first core feature of the WildlifeDatasets toolkit allows downloading, extracting, and pre-processing all 31 publicly available wildlife datasets[2] (refer to Table 1) in a unified format using just a few lines of Python code. For reference, see provided code snippet in Figure 2. 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.

3.2. 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 [38].

```
#Import wildlife-datasets Library
from wildlife_datasets import datasets, splits

#Download dataset
datasets.ATRW.get_data('data/ATRW')

#Load metadata
metadata = datasets.ATRW('data/ATRW')

#Get 80/20 training/test split
splitter = splits.ClosedSetSplit(0.8)

splitter.split(metadata.df)
>>> [<train indices>, <test indices>]
```

Figure 2. **Dataset download with WildlifeDatasets**. A code snippet showcasing easy data download, metadata load, and splitting.

Name	Year	# Images	# Identities	Timestamp	In-the-wild	Pattern	Multispecies
AAUZebraFishID [12]	2020	6672	6	✗	✗	✗	✗
AerialCattle2017 [8]	2017	46340	23	✗	✗	✓	✗
ATRW [31]	2019	5415	182	✗	✗	✓	✗
BelugaID [3]	2022	5902	788	✓	✓	✗	✗
BirdIndividualID [22]	2019	51934	50	✗	✗	✗	✓
CTai [23]	2016	4662	71	✗	✓	✗	✗
CZoo [23]	2016	2109	24	✗	✗	✗	✗
Cows2021 [24]	2021	8670	181	✓	✗	✓	✗
Drosophila [44]	2018	∼2.6M	60	✗	✗	✓	✗
FriesianCattle2015 [9]	2016	377	40	✗	✗	✓	✗
FriesianCattle2017 [8]	2017	940	89	✗	✗	✓	✗
GiraffeZebraID [40]	2017	6925	2056	✓	✓	✓	✓
Giraffes [34]	2021	1393	178	✗	✓	✓	✗
HappyWhale [13]	2022	51033	15587	✗	✓	✓	✗
HumpbackWhaleID [2]	2019	15697	5004	✗	✓	✓	✗
HyenaID2022 [48]	2022	3129	256	✗	✓	✓	✗
IPanda50 [51]	2021	6874	50	✗	✗	✓	✗
LeopardID2022 [48]	2022	6806	430	✗	✓	✓	✗
LionData [18]	2020	750	94	✗	✓	✓	✗
MacaqueFaces [54]	2018	6280	34	✓	✗	✗	✗
NDD20 [47]	2020	2657	82	✗	✗	✓	✗
NOAARightWhale [1]	2015	4544	447	✗	✓	✗	✗
NyalaData [18]	2020	1942	237	✗	✓	✓	✗
OpenCows2020 [7]	2020	4736	46	✗	✗	✓	✗
SealID [36]	2022	2080	57	✗	✓	✓	✗
SeaTurtleID [38]	2022	7774	400	✓	✓	✓	✗
SeaTurtleID2022 [5]	2024	8729	438	✓	✓	✓	✗
SMALST [57]	2019	12850	10	✗	✗	✓	✗
StripeSpotter [30]	2011	820	45	✗	✓	✓	✗
WhaleSharkID [27]	2020	7693	543	✓	✓	✓	✗
ZindiTurtleRecall [4]	2022	12803	2265	✗	✓	✓	✗

Table 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.

[1] Both libraries are available online on GitHub.

[2] Based on our research at the end of September 2023.

3.3. 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 re-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.

Local descriptors: Due to extensive utilization among ecologists and state-of-the-art performance in animal re-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 deep learning-based Superpoint descriptors. Besides, we have implemented a matching algorithm that uses local descriptors using contemporary insights and knowledge. Leveraging GPU implementation (FAISS [28]) for nearest neighbor search, we have eliminated the necessity for using approximate neighbors. This alleviates the time-complexity concerns raised by authors of the Hotspotter tool.

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 re-identification datasets. We have accomplished this by integrating the Timm library [53], which includes state-of-the-art CNN- and transformer-based architectures, e.g., ConvNeXt [33], ResNext [55], ViT [19], and Swin [32]. This integration enables both the feature extraction and the fine-tuning of models on the wildlife re-identification datasets.

MegaDescriptor: Furthermore, we provide the first-ever foundation model for individual re-identification within a wide range of species – MegaDescriptor – 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, see Figure 3 for reference.

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. For reference, see Figure 4.

```python
import timm
import torchvision.transforms as T
from PIL import Image

# Load model from Huggingface Hub
model = timm.create_model(
    "hf-hub:BVRA/MegaDescriptor-L-384",
    pretrained=True)

model = model.eval()

# Load expected image transformations
transforms = T.Compose([T.Resize(224),
                        T.ToTensor(),
                        T.Normalize(
                            [0.5, 0.5, 0.5],
                            [0.5, 0.5, 0.5])])

# Load/feed-forward image to MegaDescriptor
image = Image.open("./test_image.png")

output = model(transforms(image).unsqueeze(0))
```

Figure 3. **Inference with MegaDescriptor**. A code snippet showcasing inference with the pre-trained MegaDescriptor model.

```python
from wildlife_tools.data import FeatureDatabase
from wildlife_tools.inference import KnnMatcher

# Load database of image features
database = FeatureDatabase.from_file(
    "database_file"
)

# Extract features from query image
image = Image.open("./query_image.png")
query = model(transforms(image).unsqueeze(0))

# Find nearest match in database
matcher = KnnMatcher(database)
matcher([query])
>>> ["id_george"]
```

Figure 4. **Matching with WildlifeDatasets**. A code snippet showcasing accessible matching with already loaded pre-trained model.

3.4. 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, enabling its expansion and improvement. 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.

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

4. MegaDescriptor – Methodology

Wildlife re-identification is usually formulated as a closed-set classification problem, where the task is to assign identities from a predetermined set of known identities to given unseen images. Our setting draws inspiration from real-life applications, where animal ecologists compare a reference image set (i.e., a database of known identities) with a query image set (i.e., newly acquired images) to determine the identities of the individuals in new images. In the search for the best suitable methods for the MegaDescriptor, we follow up on existing literature [16, 31, 34, 41] and focus on local descriptors and metric Learning. We evaluate all the ablation studies over 29 datasets[4] provided through the WildlifeDataset toolkit.

4.1. Local features approaches

Drawing inspiration from the success of local descriptors in existing wildlife re-identification tools [21, 41], 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.

4.2. Metric learning approaches

Following the recent progress in human and vehicle re-id [14, 35, 56], we select two metric learning methods for our ablation studies – Arcface [17] and Triplet loss [46] – which both learn a representation function that maps objects into a deep embedding space. The distance in the embedded space should depend on visual similarity between all identities, i.e., samples of the same individual are close, and different identities are far away. CNN- or transformer-based architectures are usually used as feature extractors.

The **Triplet loss** [26, 46] involves training the model using triplets (x_a, x_p, x_n), where the anchor x_a shares the same label as the positive x_p, but a different label from the negative x_n. The loss learns embedding where the distance between x_a and x_p is small while distance between x_a and x_n is large such that the former pair should be distant to latter by at least a margin m. Learning can be further improved by a suitable triplet selection strategy, which we consider as a hyperparameter. We consider 'all' to include all valid triplets in batch, 'hard' for triplets where x_n is closer to the x_a than the x_p and 'semi' to select triplets where x_n is further from the x_a than the x_p.

The **ArcFace loss** [17] enhances the standard softmax loss by introducing an angular margin m to improve the discriminative capabilities of the learned embeddings. The embeddings are both normalized and scaled, which places them on a hypersphere with a radius of s. Value of scale s is selected as hyperparameter.

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 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 [35]. Therefore, we perform an exhaustive hyperparameters 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 2.

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

Table 2. **Grid-search setup**. Selected hyperparameters and their appropriate values for ArcFace and Triplet approaches.

[4] We avoided Drosophila (low complexity and high image number) and SeaTurleID2022 [5] due to its big overlap with SeaTurleIDHeads [38].

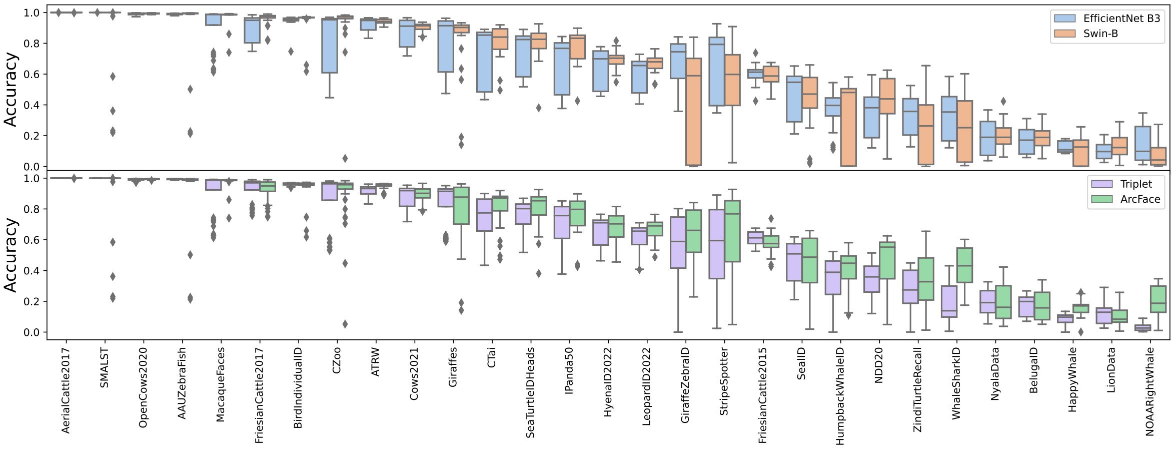


Figure 5. **Ablation of the backbone architecture and metric learning method**. We compare two backbones – Swin-B and EfficientNet-B3 – and Triplet / ArcFace methods on all available animal re-id datasets. In most cases, the Swin-B with ArcFace maintains competitive or better performance than EfficientNet-B3 and Triplet.

5. 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., first-ever foundation model for animal re-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.

5.1. 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 re-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.

5.2. 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[5]. 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 6 as a boxplot, where each box consists of 29 values.

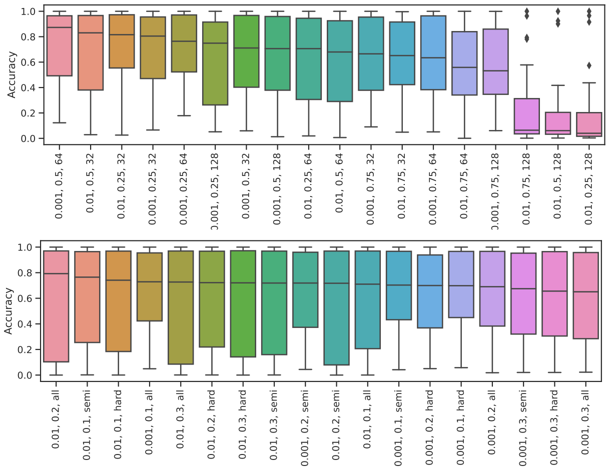


Figure 6. **Ablation of hyperparameters search**. We display performance for all settings as a boxplot combining accuracy from all 29 datasets. ArcFace (Top) and Triplet loss (Bottom).

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

5.3. 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 re-identification, even though they are rarely used over SIFT descriptors in the literature. A detailed comparison is provided in Table 3. 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.

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 3. **Ablation of animal re-id 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.

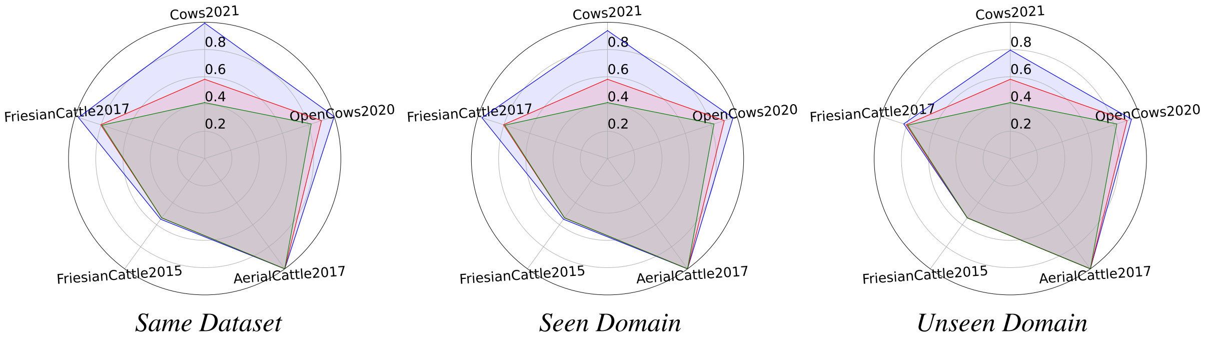
6. Performance evaluation

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 re-id 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-trained models. The proposed MegaDescriptor with Swin-L/p4-w12-384 backbone performs consistently on all datasets and outperforms all methods in 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 re-identification.

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 4. **Animal re-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 7. **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.

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

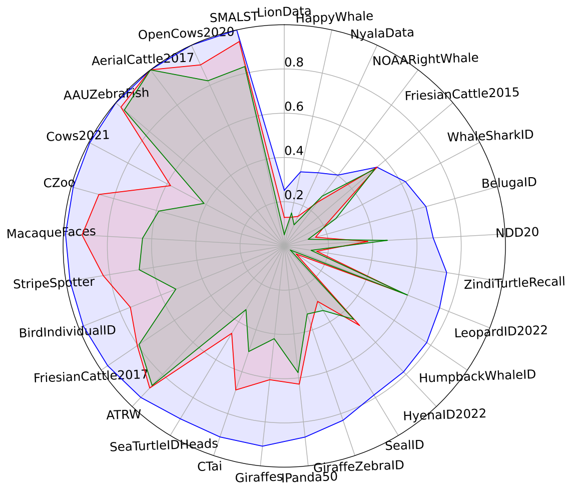


Figure 8. ***Pre-trained* models performance evaluation.** We compare DINOv2 (ViT-L/p14-518), CLIP (ViT-L/p14-336), and MegaDescriptor-L (Swin-L/p4-w12-384) on 29 selected datasets.

The MegaDescriptor-B, compared with a CLIP and DINOv2, yields significantly better or competitive performance (see Figure 7). 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. 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 re-identification in any animal datasets, and subsequently transferring this knowledge to the re-identification of the cattle.

7. Conclusion

We have introduced the WildlifeDatasets toolkit, an open-source, user-friendly library that provides (i) convenient access and manipulation of all publicly available wildlife datasets for individual re-identification, (ii) access to a variety of state-of-the-art models for animal re-identification, and (iii) simple API that allows inference and matching over new datasets. Besides, we have provided the most comprehensive experimental comparison of these datasets and essential methods in wildlife re-identification using local descriptors and deep learning approaches. Using insights from our ablation studies led to the creation of a MegaDescriptor, the first-ever foundation model for animal re-identification, which delivers state-of-the-art performance on a wide range of species. We anticipate that this toolkit will be widely used by both computer vision scientists and ecologists interested in wildlife re-identification and will significantly facilitate progress in this field.

This research was supported by the Czech Science Foundation (GA CR), project No. GA22-32620S and by the Technology Agency of the Czech Republic, project No. SS05010008. Computational resources were provided by the OP VVV project "Research Center for Informatics" (No. CZ.02.1.01/0.0/0.0/16_019/0000765).

References

[1] Right whale recognition, 2015. 1, 3

[2] Humpback whale identification, 2019. 1, 3

[3] Beluga ID 2022, 2022. 1, 3

[4] Turtle recall: Conservation challenge, 2022. 1, 3

[5] Lukáš Adam, Vojtěch Čermák, Kostas Papafitsoros, and Lukas Picek. SeaTurtleID2022: A long-span dataset for reliable sea turtle re-identification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024. 3, 5

[6] Carlos JR Anderson, Niels Da Vitoria Lobo, James D Roth, and Jane M Waterman. Computer-aided photo-identification system with an application to polar bears based on whisker spot patterns. *Journal of Mammalogy*, 91(6):1350–1359, 2010. 2

[7] William Andrew, Jing Gao, Siobhan Mullan, Neill Campbell, Andrew W Dowsey, and Tilo Burghardt. Visual identification of individual holstein-friesian cattle via deep metric learning. *Computers and Electronics in Agriculture*, 185:106133, 2021. 3

[8] 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*, pages 2850–2859, 2017. 3

[9] William Andrew, Sion Hannuna, Neill Campbell, and Tilo Burghardt. 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)*, pages 484–488. IEEE, 2016. 2, 3

[10] Anka Bedetti, Cathy Greyling, Barry Paul, Jennifer Blondeau, Amy Clark, Hannah Malin, Jackie Horne, Ronny Makukule, Jessica Wilmot, Tammy Eggeling, et al. System for elephant ear-pattern knowledge (seek) to identify individual african elephants. *Pachyderm*, 61:63–77, 2020. 2

[11] Douglas T Bolger, Thomas A Morrison, Bennet Vance, Derek Lee, and Hany Farid. A computer-assisted system for photographic mark–recapture analysis. *Methods in Ecology and Evolution*, 3(5):813–822, 2012. 2

[12] Joakim Bruslund Haurum, Anastasija Karpova, Malte Pedersen, Stefan Hein Bengtson, and Thomas B Moeslund. Re-identification of zebrafish using metric learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 1–11, 2020. 2, 3

[13] Ted Cheeseman, Tory Johnson, Ken Southerland, and Noah Muldavin. Happywhale: Globalizing marine mammal photo identification via a citizen science web platform. *Happywhale, Santa Cruz*, 2017. 1, 3

[14] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2017. 5

[15] Jonathan P Crall, Charles V Stewart, Tanya Y Berger-Wolf, Daniel I Rubenstein, and Siva R Sundaresan. Hotspotter-patterned species instance recognition. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 230–237. IEEE, 2013. 2

[16] Debayan Deb, Susan Wiper, Sixue Gong, Yichun Shi, Cori Tymoszek, Alison Fletcher, and Anil K Jain. Face recognition: Primates in the wild. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–10. IEEE, 2018. 2, 5

[17] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 5

[18] 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)*, pages 224–228. IEEE, 2020. 1, 3

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[20] 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. *Ecology and Evolution*, 5(1):141–151, 2015. 2

[21] Stephen G Dunbar, Edward C Anger, Jason R Parham, Colin Kingen, Marsha K Wright, Christian T Hayes, Shahnaj Safi, Jason Holmberg, Lidia Salinas, and Dustin S Baumbach. Hotspotter: Using a computer-driven photo-id application to identify sea turtles. *Journal of Experimental Marine Biology and Ecology*, 535:151490, 2021. 1, 2, 5

[22] André C Ferreira, Liliana R Silva, Francesco Renna, Hanja B Brandl, Julien P Renoult, Damien R Farine, Rita Covas, and Claire Doutrelant. Deep learning-based methods for individual recognition in small birds. *Methods in Ecology and Evolution*, 11(9):1072–1085, 2020. 3

[23] Alexander Freytag, Erik Rodner, Marcel Simon, Alexander Loos, Hjalmar S Kühl, and Joachim Denzler. Chimpanzee faces in the wild: Log-Euclidean CNNs for predicting identities and attributes of primates. In *German Conference on Pattern Recognition*, pages 51–63. Springer, 2016. 1, 3

[24] Jing Gao, Tilo Burghardt, William Andrew, Andrew W Dowsey, and Neill W Campbell. Towards self-supervision for video identification of individual holstein-friesian cattle: The Cows2021 dataset. *arXiv preprint arXiv:2105.01938*, 2021. 3

[25] Andrew Gilman, Krista Hupman, Karen A Stockin, and Matthew DM Pawley. Computer-assisted recognition of dolphin individuals using dorsal fin pigmentations. In *2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2016. 2

[26] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 5

[27] Jason Holmberg, Bradley Norman, and Zaven Arzoumanian. Estimating population size, structure, and residency time for whale sharks Rhincodon typus through collaborative photo-identification. *Endangered Species Research*, 7(1):39–53, 2009. 3

[28] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 4

[29] Marcella J Kelly. Computer-aided photograph matching in studies using individual identification: an example from Serengeti cheetahs. *Journal of Mammalogy*, 82(2):440–449, 2001. 2

[30] Mayank Lahiri, Chayant Tantipathananandh, Rosemary Warungu, Daniel I Rubenstein, and Tanya Y Berger-Wolf. Biometric animal databases from field photographs: identification of individual zebra in the wild. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 1–8, 2011. 3

[31] Shuyuan Li, Jianguo Li, Hanlin Tang, Rui Qian, and Weiyao Lin. ATRW: A benchmark for amur tiger re-identification in the wild. *arXiv preprint arXiv:1906.05586*, 2019. 1, 2, 3, 5

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 4

[33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 4

[34] Vincent Miele, Gaspard Dussert, Bruno Spataro, Simon Chamaillé-Jammes, Dominique Allainé, and Christophe Bonenfant. Revisiting animal photo-identification using deep metric learning and network analysis. *Methods in Ecology and Evolution*, 12(5):863–873, 2021. 2, 3, 5

[35] 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*, pages 681–699. Springer, 2020. 5

[36] Ekaterina Nepovinnykh, Tuomas Eerola, Vincent Biard, Piia Mutka, Marja Niemi, Heikki Kälviäinen, and Mervi Kunnasranta. SealID: Saimaa ringed seal re-identification dataset. *arXiv preprint arXiv:2206.02260*, 2022. 3

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[38] Kostas Papafitsoros, Lukáš Adam, Vojtěch Čermák, and Lukáš Picek. SeaTurtleID: A novel long-span dataset highlighting the importance of timestamps in wildlife re-identification. *arXiv preprint arXiv:2211.10307*, 2022. 3, 5

[39] Kostas Papafitsoros, Aliki Panagopoulou, and Gail Schofield. Social media reveals consistently disproportionate tourism pressure on a threatened marine vertebrate. *Animal Conservation*, 24(4):568–579, 2021. 1

[40] Jason Remington Parham, Jonathan Crall, Charles Stewart, Tanya Berger-Wolf, and Daniel Rubenstein. Animal population censusing at scale with citizen science and photographic identification. In *2017 AAAI Spring Symposium Series*, 2017. 3

[41] Malte Pedersen, Joakim Bruslund Haurum, Thomas B Moeslund, and Marianne Nyegaard. Re-identification of giant sunfish using keypoint matching. In *Proceedings of the Northern Lights Deep Learning Workshop*, volume 3, 2022. 5

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[43] Vito Renò, Giovanni Dimauro, G Labate, Ettore Stella, Carmelo Fanizza, Giulia Cipriano, Roberto Carlucci, and Rosalia Maglietta. A SIFT-based software system for the photo-identification of the Risso's dolphin. *Ecological informatics*, 50:95–101, 2019. 2

[44] Jonathan Schneider, Nihal Murali, Graham W Taylor, and Joel D Levine. Can Drosophila melanogaster tell who's who? *PloS one*, 13(10):e0205043, 2018. 3

[45] Gail Schofield, Kostas Papafitsoros, Chloe Chapman, Akanksha Shah, Lucy Westover, Liam CD Dickson, and Kostas A Katselidis. More aggressive sea turtles win fights over foraging resources independent of body size and years of presence. *Animal Behaviour*, 190:209–219, 2022. 1

[46] 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*, pages 815–823, 2015. 5

[47] Cameron Trotter, Georgia Atkinson, Matt Sharpe, Kirsten Richardson, A Stephen McGough, Nick Wright, Ben Burville, and Per Berggren. NDD20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation. *arXiv preprint arXiv:2005.13359*, 2020. 1, 3

[48] Botswana Predator Conservation Trust. Panthera pardus csv custom export, 2022. 1, 3

[49] Masataka Ueno, Ryosuke Kabata, Hidetaka Hayashi, Kazunori Terada, and Kazunori Yamada. Automatic individual recognition of Japanese macaques (Macaca fuscata) from sequential images. *Ethology*, 128(5):461–470, 2022. 2

[50] Maxime Vidal, Nathan Wolf, Beth Rosenberg, Bradley P Harris, and Alexander Mathis. Perspectives on individual animal identification from biology and computer vision. *Integrative and Comparative Biology*, 61(3):900–916, 2021. 1

[51] Le Wang, Rizhi Ding, Yuanhao Zhai, Qilin Zhang, Wei Tang, Nanning Zheng, and Gang Hua. Giant panda identification. *IEEE Transactions on Image Processing*, 30:2837–2849, 2021. 3

[52] Hendrik Weideman, Chuck Stewart, Jason Parham, Jason Holmberg, Kiirsten Flynn, John Calambokidis, D Barry Paul, Anka Bedetti, Michelle Henley, Frank Pope, and Jerenimo Lepirei. 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*, pages 1276–1285, 2020. 2

[53] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 4

[54] Claire L Witham. Automated face recognition of rhesus macaques. *Journal of Neuroscience Methods*, 300:157–165, 2018. 1, 3

[55] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, Honolulu, 2017. 4

[56] Cheng Yan, Guansong Pang, Xiao Bai, Changhong Liu, Xin Ning, Lin Gu, and Jun Zhou. Beyond triplet loss: person re-identification with fine-grained difference-aware pairwise loss. *IEEE Transactions on Multimedia*, 24:1665–1677, 2021. 5

[57] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J Black. 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*, pages 5359–5368, 2019. 1, 3