
























**SciPy 2025**

July 7 - July 13, 2025

Proceedings of the 24<sup>th</sup>  
Python in Science Conference  
ISSN: 2575-9752

# Zamba: Computer vision for wildlife conservation

Emily Dorne<sup>1</sup> , Jay Qi<sup>1</sup> , Peter Bull<sup>1</sup> , Colleen Stephens<sup>2</sup> , Mattia Bessone<sup>2,3,4</sup> , Benjamin Debetencourt<sup>5,2</sup> , Barbara Fruth<sup>6,7</sup> , David Morgan<sup>8</sup> , Meredith S. Palmer<sup>9</sup> , Crickette Sanz<sup>10,11</sup> , Janika Wendeferer<sup>12,13</sup> , Catherine Crockford<sup>14,15</sup> , Tobias Deschner<sup>16,2</sup> , Kevin E. Langergraber<sup>17,18</sup> , Alex K. Piel<sup>19,20,21</sup> , Martha Robbins<sup>22</sup> , Volker Sommer<sup>19,23</sup> , Fiona A. Stewart<sup>19,20,21</sup> , Roman M. Wittig<sup>14,15</sup> , Klaus Zuberbuehler<sup>24,25</sup> , Hjalmar S. Kühl<sup>26,27</sup> , and Mimi Arandjelovic<sup>22,28</sup> 

<sup>1</sup>DrivenData, <sup>2</sup>Max Planck Institute for Evolutionary Anthropology, <sup>3</sup>Centre for the Advanced Study of Collective Behaviour, University of Konstanz, <sup>4</sup>Department of Animal Societies, Max Planck Institute of Animal Behavior, <sup>5</sup>Wild Chimpanzee Foundation, <sup>6</sup>Department for the Ecology of Animal Societies, Max-Planck Institute of Animal Behavior, <sup>7</sup>Centre for Research and Conservation/KMDA, <sup>8</sup>Lester E. Fisher Center for the Study and Conservation of Apes, Lincoln Park Zoo, <sup>9</sup>Center for Biodiversity and Global Change, Yale University, <sup>10</sup>Department of Anthropology, Washington University in Saint Louis, <sup>11</sup>Congo Program, Wildlife Conservation Society, <sup>12</sup>Universität Hamburg, <sup>13</sup>WWF Central African Republic, <sup>14</sup>Ape Social Mind Lab, Institute for Cognitive Sciences Marc Jeannerod (UMR 5229), CNRS and University of Lyon 1, <sup>15</sup>Tai Chimpanzee Project, Centre Suisse de Recherches Scientifiques, <sup>16</sup>Institute of Cognitive Science, University of Osnabrück, <sup>17</sup>School of Human Evolution and Social Change, Arizona State University, <sup>18</sup>Institute of Human Origins, <sup>19</sup>Department of Anthropology, University College London, <sup>20</sup>Department of Human Origins, Max Planck Institute for Evolutionary Anthropology, <sup>21</sup>GMERC, LTD. (Greater Mahale Ecosystem Research and Conservation Project), <sup>22</sup>Department of Primate Behavior and Evolution, Max Planck Institute for Evolutionary Anthropology, <sup>23</sup>Gashaka Primate Project, <sup>24</sup>Institut de Biologie, Université de Neuchâtel, <sup>25</sup>School of Psychology and Neuroscience, University of St Andrews, <sup>26</sup>Senckenberg Museum of Natural History Görlitz, <sup>27</sup>International Institute Zittau, Technische Universität Dresden, <sup>28</sup>German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig


## Abstract

Camera traps are indispensable tools for wildlife conservation, but the vast volume of data they generate creates a significant bottleneck in manual processing. We present **Zamba**, an open-source Python package that leverages machine learning to streamline camera trap data analysis. Unlike most existing tools that support only image data, Zamba enables species classification for both still images and videos, as well as depth estimation from videos. It also supports custom model training on user-provided data, allowing adaptation to new species and geographic regions not covered by pretrained models. Zamba powers **Zamba Cloud**, a no-code web application that makes these capabilities accessible to non-programmers. By enabling conservationists to efficiently analyze large datasets and train models tailored to their specific ecological contexts, Zamba advances wildlife monitoring, research, and evidence-based conservation decision-making.

**Keywords** camera traps, computer vision, machine learning, wildlife conservation, species classification, depth estimation

**Published** Jul 10, 2025

**Correspondence to**  
Emily Dorne  
[emily@drivendata.org](mailto:emily@drivendata.org)

**Open Access** 

Copyright © 2025 Dorne *et al.*. This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license, which enables reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator.

## 1. INTRODUCTION

Camera traps are automated camera systems typically triggered by motion, heat, or timers [1]. They are a widely used tool for wildlife monitoring in both research and conservation. However, they generate an immense volume of image and video data due to the nature of the data collection and scale of their deployment. It takes the valuable time of teams of experts, or thousands of citizen scientists, to manually process this data and identify videos and images of interest [2].

Automated computer vision systems offer a natural solution to assist with this data processing. Yet, two key gaps remain in the ecosystem for wildlife camera trap data. First, available domain-specific tools only support processing image data, not video [3], and it is not computationally feasible to predict on all frames in a video as if they were images. Moreover, models that process video natively can leverage inter-frame information—such as motion—that is otherwise lost. Second, pretrained models fail to capture the wide range of environments and use cases for which camera traps are deployed. Wildlife monitoring projects vary widely in habitat, focal species, deployment context, and taxonomic resolution, necessitating adaptable models that can be fine-tuned to specific contexts.

We present [Zamba](#) [4], an open source Python package for using machine learning to process camera trap data. Zamba supports species classification for both videos and still images. Zamba also supports custom model training on user-provided camera trap data containing new species and new geographies not in the included pretrained models. In addition, Zamba includes inference-only video processing pipelines for depth estimation. Zamba also underlies [Zamba Cloud](#), which is an accessible no-code web application that runs the python package on managed cloud resources. Zamba Cloud bridges a critical gap between the technical feasibility of training models and the practical ability for conservationists — who are often not programmers — to use them.

By enabling conservationists to efficiently process large datasets and train custom models suited to their unique ecological contexts and research questions, Zamba accelerates wildlife monitoring, research, and evidence-based conservation decision-making.

This paper provides an overview of Zamba’s contributions to the conservation community and details its underlying methodology. The motivation section illustrates the value of camera trap data for conservation efforts as well as the current bottlenecks. The development section reviews Zamba’s history. The following three sections detail the methods and performance for the video species classification, image species classification, and depth estimation models. We then discuss how Zamba Cloud provides an accessible no-code way to access Zamba’s classification capabilities. The discussion section concludes with guidance on common workflows.

## 2. MOTIVATION

Camera traps hold immense potential for conservationists as they are a noninvasive way of monitoring wildlife [5]. They enable the measurement of species occupancy [6], relative [7] or absolute abundance [8], population trends over time [9], patterns in animal behavior [3], [10], and more. They are also one of the few tools for monitoring multiple species simultaneously [11].

As simple sensors, camera traps do not automatically label or filter the species they observe. Manual review and labeling represent a critical bottleneck, consuming vast amounts of resources that could otherwise support direct conservation actions [2], [12].

One of the most common needs is filtering out “blank” images and videos, i.e., those that do not contain an animal but rather were a false trigger, potentially caused by wind, rain,

or changes in light. These “blanks” account for up to 70% of captured images [13] in some datasets. Estimating animal distances in camera trap footage currently entails an extremely manual, time-intensive process. Automated distance extraction is estimated to be 21 times faster than manual labelling [14].

While more and more conservation organizations and researchers are collecting and storing camera trap data, many are not yet able to effectively use that data for decision-making. The people-power needed to view and annotate camera trap data is prohibitively expensive in terms of time and money, which results in organizations not using camera traps; data accumulating on hard drives; or long delays in turn-around while teams sift through thousands of incoming images and videos every month.

Machine learning (ML) models have shown significant promise in automating species identification from camera trap imagery [3]. Several software tools that leverage ML have gained widespread adoption in recent years, including MegaDetector [13], Wildlife Insights [15], and Addax AI (formerly EcoAssist) [16]. These tools have advanced the application of pretrained computer vision models for animal detection and species classification particularly for camera trap image data.

Accurate, accessible, and automated species detection enables improved monitoring of animal populations and evaluation of conservation impacts on species abundance. Faster processing of camera trap data allows conservationists to conduct these assessments in weeks instead of years. This not only supports timely evaluations and interventions—allowing for adaptive management if efforts are not proving effective—but also enables the collection of more data from a broader range of locations. It also allows conservationists to direct their time toward more complex secondary analyses and making evidence-based conservation decisions.

### 3. DEVELOPMENT HISTORY

Zamba’s development has its origins in online machine learning challenges hosted by DrivenData in partnership with wildlife conservation experts—in particular from the Max Planck Institute for Evolutionary Anthropology (MPI-EVA). Machine learning competitions have proven effective at harnessing community-driven innovation and enabling the rapid testing of thousands of approaches to solving a problem [17]. Zamba, named after the word for “forest” in Lingala<sup>1</sup>, was created to make these advances more accessible to ecologists and conservationists.

#### 2017: Pri-matrix Factorization Challenge and Zamba v1

In the [Pri-matrix Factorization Challenge](#) [18], hosted by DrivenData and MPI-EVA, over 320 participants competed to develop accurate species detection models for camera trap video data. The competition used a unique dataset created through the Chimp&See Zooniverse project [2], where thousands of citizen scientists manually labeled camera trap videos. Their efforts produced nearly 2,000 hours of annotated footage from the Chimp&See database.<sup>2</sup> The winning algorithm achieved 96% accuracy in detecting wildlife presence and 99% average accuracy in classifying species across 23 label classes [19] and formed the foundation of the initial version<sup>3</sup> of Zamba, available through a command-line interface (CLI).

<sup>1</sup>[Lingala](#) is a Bantu language spoken widely across Central Africa, particularly in the Democratic Republic of the Congo and the Republic of the Congo.

<sup>2</sup>Since this data was non-expert labeled, certain thresholds on how many user annotations were required to accept a label as well as thresholds related to percentages of user agreement were applied to go from raw annotations to a well-labeled dataset.

<sup>3</sup>Since Zamba’s origins in 2017, computer vision techniques have evolved significantly. The original algorithm implemented in Zamba v1 used an ensemble of five models with architectures based on convolutional neural networks (CNNs)—ImageNet—ResNet50, ResNet152 [20], InceptionV3 [21], Xception

## 2018: Zamba Cloud

Zamba was further developed into a web application—Zamba Cloud—to promote wider accessibility and adoption. Zamba Cloud enables users to run Zamba on high-performance, cloud-based infrastructure through an easy-to-use web interface.

## 2021: Zamba v2

Zamba underwent a major re-architecture to leverage recent advances in computer vision, released as Zamba v2.0.0. Its models were also retrained on an expanded dataset of 250,000 labeled camera trap videos.<sup>4</sup> This update also introduced support for fine-tuning custom models on videos and incorporated a smarter frame selection strategy using a newly trained machine learning model called [MegadetectorLite](#).

## 2021: Deep Chimpact Challenge and depth estimation

DrivenData partnered again with the Max Planck Institute for Evolutionary Anthropology and the Wild Chimpanzee Foundation to host the [Deep Chimpact: Depth Estimation for Wildlife Conservation](#) challenge [25]. More than 300 participants competed to develop models for monocular depth estimation—inferring the distance between camera traps and wildlife appearing at various moments in video footage. In 2022, the second-place model from this challenge was integrated into Zamba as an inference module.

## 2024–2025: Species classification for images

In March 2025, species classification for images was added in Zamba v2.6.0 and Zamba Cloud. This major update, supported by the WILDLABS Awards 2024, focused on enabling non-programmer users to train custom image classification models through Zamba Cloud’s no-code interface. The work involved aggregating diverse image datasets, experimenting with model architectures, and training a robust base model spanning 178 taxonomic label classes.

# 4. CLASSIFYING SPECIES IN VIDEOS

Camera trap videos are especially valuable for researchers studying animal behavior, as they capture rich information including sounds, movements, tool use, and interactions between individuals [26], [27], [28]. Videos also provide multiple views of the same individual, which can aid in identifying characteristics such as sex, size, and age [2], [29], [30]. It can also enable individual identification, which is useful for capture-recapture studies [31].

However, video classification is more challenging than image classification. It demands more complex model architectures, increased computational resources such as graphics processing units (GPUs), and the ability to manage and process datasets often measured in terabytes. Zamba initially focused on videos in order to address the gap in the tooling ecosystem, as other automated species classification tools only supported still images.

In the sections that follow, we describe Zamba’s approach to species classification in videos. This includes the core methodologies, the available pretrained models and features, and performance metrics based on evaluation on a holdout set.

---

[22], and Inception-ResNetV2 [23]. Base models trained on ImageNet were fine-tuned. Then, class probabilities for 32 frames per clip were aggregated into summary statistics and passed to a second-level XGBoost [24] model, which generated the final species prediction. [19] This model was replaced in 2021 with the release of Zamba v2 so is not discussed in more depth here.

<sup>4</sup>This retraining update also addressed artifacts from the original competition. For example, the competition’s lack of site-aware splitting contributed to inflated performance metrics by allowing models to learn location-specific features from backgrounds. Additionally, labeling errors stemming from crowd-sourced annotations were corrected using a curated set of 125,000 expert-labeled videos, which significantly improved performance for certain species.

## 4.1. Methods

### 4.1.1. Training data

A large and diverse dataset is one of the most important factors influencing model accuracy for camera trap species classification. Zamba’s video models were trained on a dataset of 250,000 labeled camera trap videos. Half of this data was collected and annotated by trained ecologists, while the other half consists of high-confidence labels generated by citizen scientists through the Chimp&See platform. The training data comes from camera trap deployments in the following countries: Cameroon, Central African Republic, Democratic Republic of the Congo, Gabon, Guinea, Liberia, Mozambique, Nigeria, Republic of the Congo, Senegal, Tanzania, and Uganda.

**Table 1.** *Locations included in the training data for the video species classification models*

Country	Location
Cameroon	Campo Ma’an National Park
	Korup National Park
Central African Republic	Dzanga-Sangha Protected Area
Côte d’Ivoire	GEPRENAF-Comoé
	Djouroutou
	Taï National Park
Democratic Republic of the Congo	Bili-Uele Protected Area
	Salonga National Park
Gabon	Loango National Park
	Lopé National Park
Guinea	Bakoun Classified Forest
	Moyen-Bafing National Park
Liberia	East Nimba Nature Reserve
	Grebo-Krahn National Park
	Sapo National Park
Mozambique	Gorongosa National Park
Nigeria	Gashaka-Gumti National Park
Republic of the Congo	Conkouati-Douli National Park
	Nouabalé-Ndoki National Park
Senegal	Kayan
Tanzania	Grumeti Game Reserve
	Ugalla River National Park
Uganda	Budongo Forest Reserve
	Bwindi Forest National Park
	Ngogo, Kibale National Park

A carefully considered train–test split strategy is critical for camera trap data. Because each camera remains fixed in place, many videos share identical backgrounds. Without careful splitting, models risk overfitting to site-specific visual features rather than generalizing to animal appearance. To address this, Zamba uses a site-aware split strategy, where all videos from a single camera location are placed entirely in either the training or test set.

Species labels in the dataset were standardized into 32 output classes<sup>5</sup> by manually grouping vernacular species names used in expert-labeled videos into a fixed set of target classes. This aggregation provided enough training examples per class while preserving a level of taxonomic granularity useful to most users out of the box. Although the 32 classes cover only a small portion of species studied in conservation research worldwide, Zamba also supports training custom models to cover additional species, habitats, and taxonomic ranks. See “[Custom model training](#)” for further discussion.

**Table 2.** Example vernacular names that were all mapped to the target class *antelope\_duiker*

antelope spp.	bongo	bushbuck	dikdik	duiker
eland	gazelle	grantsgazelle	hartebeest	impala
jentink’s duiker	kudu	maxwell’s duiker	nyala	ogilby’s duiker
oribi	peter’s duiker	red duiker	reedbuck	royal antelope
sable	sitatunga	thompsonsgazelle	topi	water chevrotain
waterbuck	wildebeest	white bellied duiker	yellow backed duiker	zebra duiker

An additional training dataset of European wildlife was used to fine-tune a model specialized for European species. This dataset includes camera trap videos from Hintenteiche bei Biesenbrow, Germany [32].

#### 4.1.2. Classification model architecture

Zamba includes four pretrained species classification video models that implement one of the two architectures: `time_distributed` or `slowfast`.<sup>6</sup>

The `time_distributed` model architecture is based on EfficientNetV2 [38]. EfficientNetV2 models are convolutional neural networks designed to jointly optimize model size and training speed. EfficientNetV2 is image-native, meaning it operates on each frame individually. The model is wrapped in a `TimeDistributed` layer [39], which aggregates frame-level predictions into a single video-level prediction.

The `slowfast` model architecture uses the SlowFast [40] backbone for video classification. This model is named for its two parallel pathways: a slow pathway that processes low frame-rate inputs to capture spatial semantics, and a fast pathway that processes high frame-rate inputs to capture motion dynamics. As a video-native architecture, SlowFast models consider temporal relationships between frames, which can be especially valuable for detecting occluded animals that may only become apparent through movement.

[Table 3](#) provides an overview of the models. The `time_distributed` and `slowfast` models are trained to classify 32 species or species groups common to Central and West Africa.<sup>7</sup> The european model is trained to classify 11 common species in Western Europe.<sup>8</sup> Each model has distinct relative strengths. For example, the `slowfast` model trained on African species may perform better for small-bodied animals like rodents than the `time_distributed` African

<sup>5</sup>These output classes were selected based on the research needs of the data providers from the Max Planck Institute for Evolutionary Anthropology.

<sup>6</sup>Other architectures evaluated during development in 2021 included image-based and video-based models such as ResNet [33], R2Plus1D [34], TimeSFormer [35], X3D [36], and I3D [37].

<sup>7</sup>The label classes for the `time_distributed` and `slowfast` African species models are: aardvark, antelope\_duiker, badger, bat, bird, blank, cattle, cheetah, chimpanzee\_bonobo, civet\_genet, elephant, equid, forest\_buffalo, fox, giraffe, gorilla, hare\_rabbit, hippopotamus, hog, human, hyena, large\_flightless\_bird, leopard, lion, mongoose, monkey\_prosimian, pangolin, porcupine, reptile, rodent, small\_cat, wild\_dog\_jackal

<sup>8</sup>The label classes for the european model are: bird, blank, domestic\_cat, european\_badger, european\_beaver, european\_hare, european\_roe\_deer, north\_american\_raccoon, red\_fox, weasel, and wild\_boar



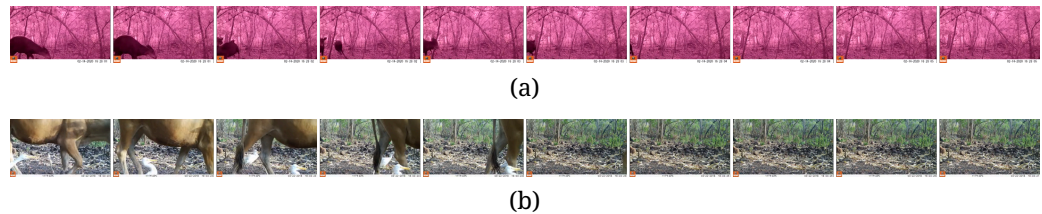
species model. There is also a european model, fine-tuned from the `time_distributed` African model, that is trained on species from non-jungle ecologies in Western Europe. All models include a `blank` class to identify videos with no animals, but the dedicated `blank_nonblank` model may provide higher accuracy for blank detection alone.

**Table 3.** Summary of Zamba’s pretrained video models

Model	Geography	Relative strengths	Architecture	Number of training videos
<code>blank_nonblank</code>	Central Africa, West Africa, and Western Europe	Just blank detection, without species classification	Image-based <code>TimeDistributedEfficientNet</code>	~263,000
<code>time_distributed</code>	Central and West Africa	Recommended species classification model for jungle ecologies	Image-based <code>TimeDistributedEfficientNet</code>	~250,000
<code>slowfast</code>	Central and West Africa	Potentially better than <code>time_distributed</code> at small species detection	Video-native <code>SlowFast</code>	~15,000
<code>european</code>	Western Europe	Trained on non-jungle ecologies	Finetuned <code>time_distributed</code> model	~13,000

#### 4.1.3. Frame selection approach

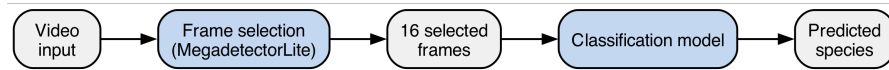
One of the key technical challenges in working with camera trap videos—rather than still images—is frame selection. For datasets that motivated the development of Zamba, videos had a frame rate of 30 frames per second and were typically about 60 seconds long. Processing every frame as an image for such videos is computationally infeasible. Furthermore, animals may only be present in a minority of recorded frames, and selecting frames without the animal degrades the performance of downstream tasks like species classification and depth estimation.



**Figure 1.** Excerpts from two 60-second example videos in which animals appear only briefly. Each excerpt shows the first 5 seconds; the remaining 55 seconds contain an unoccupied field of view.

The default frame selection method currently used by Zamba is an efficient object detection model we developed called `MegadetectorLite`. This model evaluates each frame for the likelihood that it contains an animal. By default, the top 16 frames<sup>9</sup> with the highest detection probabilities are selected and passed to either the species classification or blank detection models.

<sup>9</sup>This parameter was found empirically via crowd-sourced experimentation by participants in the Primate Factorization Challenge to be sufficient for accurately classifying videos while balancing the overall computational requirements of the pipeline. Users may choose to tune this parameter for their particular dataset and use case. See the “[Configuration and customization](#)” section.



**Figure 2.** Flow diagram of the video species classification pipeline. Gray nodes indicate data and blue nodes indicate a processing step.

Because frame selection is an upstream task, it must be fast without sacrificing too much accuracy. Larger models tend to be more accurate but are slower to run. MegadetectorLite is based on the YOLOX architecture [41] and was trained using a knowledge distillation approach [42] (also known as teacher–student training). It uses the predictions of the original MegaDetector [13] as supervisory labels. MegaDetector itself is a highly accurate, but computationally intensive, object detection model trained specifically on camera trap imagery. While MegaDetector is used directly in Zamba’s image workflows, it is too slow for frame-by-frame inference in video processing. MegadetectorLite offers a more efficient alternative with a lightweight architecture.

We experimented with two **YOLOX model variants** (yolox-nano and yolox-tiny) and two input image sizes (416 and 640 pixels). We ultimately selected yolox-tiny with an input size of 640 pixels as the best tradeoff between speed and accuracy.

MegadetectorLite was trained on over 1 million frames. To improve detection of rare species—which have historically underperformed in automated systems—the training dataset was enriched with extra frames from videos featuring less frequently observed animals, including hyenas, leopards, armadillos, reptiles, bats, giraffes, lions, cheetahs, and badgers.

## 4.2. Results

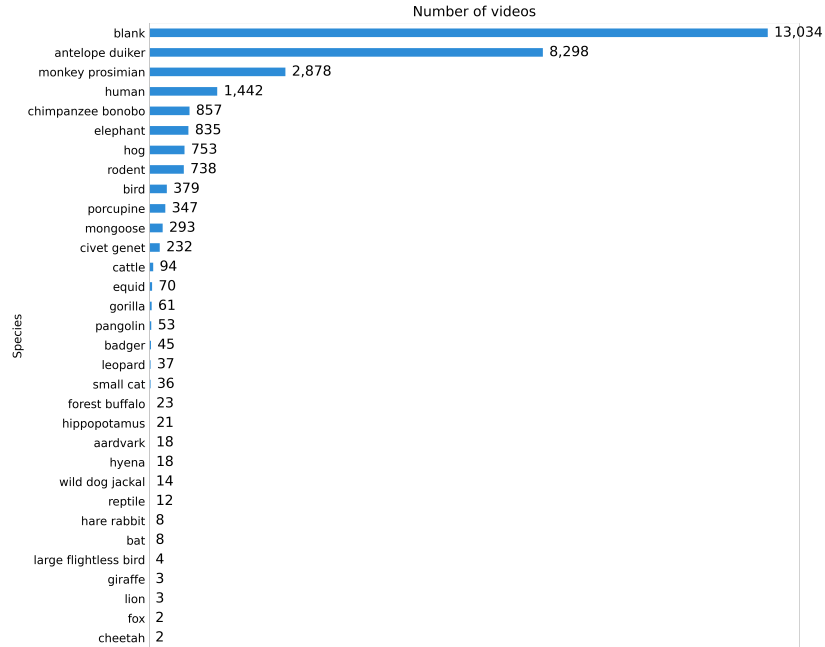
### 4.2.1. Pretrained model performance

Metrics for camera trap tasks are highly dataset-dependent and can vary widely depending on deployment context and class balance. Here, we report some illustrative evaluation metrics for Zamba’s pretrained models using holdout validation.

The holdout set comprises a random sample of labeled videos, selected on a transect-by-transect basis. That is, all videos from a given transect (camera location) are assigned entirely to either the training or holdout set. While not all use cases require this site-aware split, this is a stricter evaluation to performance on transects the model has never seen.

The holdout set includes data from all 14 countries in the training dataset, with country-level proportions roughly matching those of the complete set. [Figure 3](#) shows the distribution of videos across 30 animal species, as well as a substantial number of blank videos and some containing humans.





**Figure 3.** Distribution of label classes in video model holdout set.

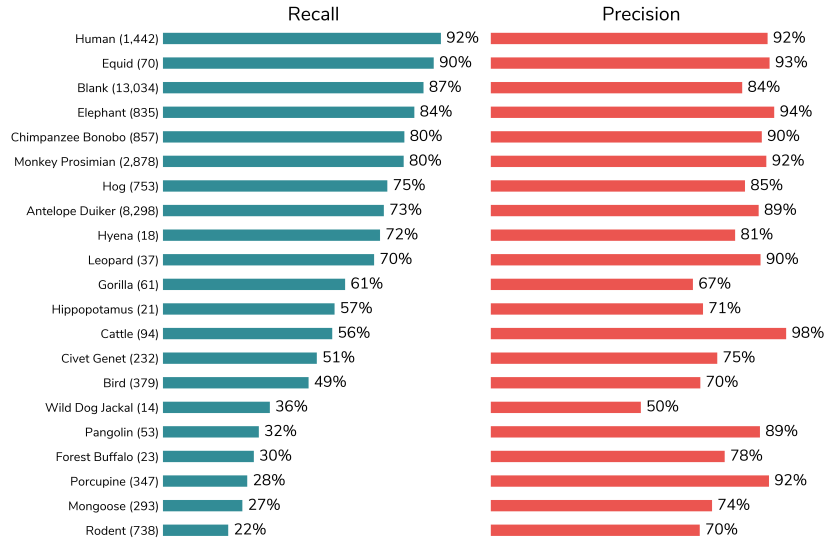
### Species classification

On the holdout set, African species `time_distributed` model achieves a top-1 accuracy<sup>10</sup> of 82% and a top-3 accuracy<sup>11</sup> of 94%. Performance varies across regions depending on species diversity. For example, in the limited sample of videos from Equatorial Guinea, only three species are represented. In this case, top-1 accuracy is 97%, significantly above the global average. In contrast, Ivory Coast videos contain 21 species, making classification more challenging; here, top-1 accuracy drops to 80%.

Figure 4 shows precision and recall by class for the holdout set. Eleven species with too few examples were excluded from this analysis. In general, recall tends to be higher for larger animals. However, even for species with lower recall, it is still feasible to use predicted probabilities to flag likely candidates for further review. For example, rodents have a recall of only 22%, but sorting videos by the model’s predicted probability for the “rodent” class remains a useful heuristic—even if other classes have higher probabilities predicted.

<sup>10</sup>Top-1 accuracy is defined as the percentage of videos where the class with the highest predicted probability is actually present.

<sup>11</sup>Top-3 accuracy is the percentage where the true class appears among the top three predicted classes.



**Figure 4.** Precision and recall by class for the holdout set.

Model performance tends to be lower for:

- Rare species with limited training data
- Small-bodied animals, especially since the video models classify the entire frame without cropping around the subject
- Heavily occluded or distant animals, or those only partially visible or present for just a few frames
- Low-quality media, such as videos that are washed out or too dark

Because of a lack of comparable wildlife camera trap tools that perform species classification for videos, it is difficult to provide a direct comparison to other results. Users are always encouraged to evaluate models on their own data. Additionally, Zamba supports fine-tuning custom models which can often be the best choice for achieving strong performance.

### Blank detection

Blank detection is an important feature in camera trap workflows due to the high prevalence of blank videos. In our holdout set, 42% of videos contained no animal activity. The `blank_nonblank` model achieved 87% recall (correctly identifies 11,288 of 13,034 blank videos) and 84% precision (13,507 videos predicted as blank, of which 11,288 were correct).

### 4.2.2. Applications to out-of-domain data

While Zamba’s pretrained models are trained to predict specific classes for species from Africa and Western Europe, it can be possible to effectively apply them to out-of-domain data that includes geographies or species that the model has never seen. In DrivenData [43], we show a case study using a dataset of 3,056 videos from New Zealand. Using the pretrained African species `time_distributed` model, the `bird` label correctly captures 77% of the takahē videos (recall) at 82% precision, which is particularly notable since the takahē is a New-Zealand-native flightless bird that is not present in the African training data. However, other New Zealand-native birds like the weka are less successfully captured by the `bird` category (44% recall). [Training custom models](#) is recommended to achieve more accurate classification on new geographies and species.

#### 4.2.3. Configuration and customization

All Zamba workflows—inference, fine-tuning, and training—support an extensive range of configuration options. For instance, processed videos are resized to configurable size, by default 240x426 pixels. Higher resolution videos will lead to superior accuracy in prediction, but will use more memory and take longer to train or predict.

Frame selection is another key configurable option. In addition to the default MegadetectorLite method, users can choose from:

- Evenly distributed frames
- Early-biased frame sampling
- Key frame extraction
- Scene-change-based selection

The number of frames selected is also customizable (default: 16).

Configurations can be provided via command-line flags or through YAML-formatted configuration files.

#### 4.2.4. Custom model training

One of Zamba’s signature features is its support for training custom models. Custom models are primarily built by fine-tuning general-purpose species classifiers and can enhance performance in specific ecological contexts—even when classifying the same species. Custom models are also essential for extending classification to new species, new habitats, or label classes at different taxonomic ranks.

Zamba supports two fine-tuning scenarios based on label compatibility:

- If the new training data uses a subset of the model’s existing classes, Zamba continues training with the existing classification head.
- If the training data includes new or different label classes, Zamba automatically replaces the model’s classification head (i.e., the final neural network layer) before continuing training.

### 5. CLASSIFYING SPECIES IN IMAGES

While Zamba began with a focus on video data, the majority of camera trap users today collect still images. Images have several practical advantages: they require less storage space, use less battery life, and are faster to transfer, review, and process. Images are often sufficient for determining presence or absence of a particular species and can be used for distance sampling [14] as well to get to abundance estimates.

There are a number of machine learning tools for processing camera trap images [3]. However, static pretrained models fail to capture the diversity in camera trap images and the range of environments in which camera traps are used around the globe.

Zamba’s extension into image support was motivated by the need to bring custom model training workflows to image-based data, building on the similar capabilities already developed for videos. The ability to train accurate models with a relatively small amount of labeled data is central to Zamba’s goal of enabling camera trap users to harness automated species classification for their unique datasets.

In the sections that follow, we outline the core methods Zamba employs for species classification in images, and present performance metrics for the general image classification model that forms the foundation for training custom models.

## 5.1. Methods

### 5.1.1. Cropping

Following the approach of T. Gadot *et al.* [44], Zamba’s image classification pipeline first applies a species-agnostic object detection model to extract bounding box crops of detected animals. The image classification model then operates solely on these cropped regions, rather than processing the full-frame image.

For object detection, Zamba uses MegaDetector [13], a widely adopted open-source model designed specifically for wildlife camera trap data. MegaDetector identifies animals, humans, and vehicles in camera trap imagery and is a standard component in many ecological processing pipelines [3]. For additional technical details and performance evaluations of MegaDetector, see J. Vélez *et al.* [3].



**Figure 5.** Flow diagram of the image species classification pipeline. Gray nodes indicate data and blue nodes indicate a processing step.

### 5.1.2. Training data

Zamba’s pretrained image classification model was trained on over 15 million annotations from over 7 million images from the [Labeled Information Library of Alexandria: Biology and Conservation](#) (LILA BC) data repository [45]. Each annotation corresponds to a cropped image of an animal, extracted using bounding boxes generated by MegaDetector [13].

To maximize geographic and ecological diversity, the training dataset aggregated 20 terrestrial camera trap datasets from LILA BC, representing a wide range of global habitats. (See [Table 4](#) for a full list of datasets.) This broad coverage was designed to produce a versatile base model well-suited for fine-tuning to a wide variety of ecological contexts.

**Table 4.** Datasets from LILA BC included in training data

Dataset	Geography	Count of original images	Count of cropped annotations
Caltech Camera Traps [46]	Southwestern United States	59,205	96,724
Channel Islands Camera Traps [47]	California, United States	125,369	239,472
Desert Lion Camera Traps [48]	Namibia	61,910	185,475
ENA24-detection [49]	Eastern North America	8,652	11,092
Idaho Camera Traps [50]	Idaho, United States	338,706	1,072,912
Island Conservation Camera Traps [51]	7 islands around the world	44,007	79,660
Missouri Camera Traps [52]	Missouri, United States	946	955

North American Camera Trap Images [53]	United States	2,705,394	7,426,839
New Zealand Trailcams [54]	New Zealand	2,109,592	2,794,859
Orinoquia Camera Traps [3]	Colombia	80,307	103,856
Snapshot Safari 2024 Expansion [55]	Africa (multiple countries)	836,522	1,949,366
Snapshot Safari Camdeboo [55]	South Africa	15,299	26,379
Snapshot Safari Enonkishu [55]	Kenya	9,049	37,252
Snapshot Safari Karoo [55]	South Africa	5,764	8,426
Snapshot Safari Kgalagadi [55]	South Africa and Botswana	2,060	2,938
Snapshot Safari Kruger [55]	South Africa	3,112	6,343
Snapshot Safari Mountain Zebra [55]	South Africa	5,535	9,333
SWG Camera Traps [56]	Vietnam and Laos	87,309	100,677
WCS Camera Traps [57]	12 countries	523,897	920,471
Wellington Camera Traps [58]	New Zealand	203,038	269,146

A key challenge in using data from these disparate sources was the lack of a unified taxonomy across datasets. Each dataset was collected independently, often for different research purposes, and the taxonomic granularity varies significantly. For example, some datasets label all birds under a generic “bird” class, while others distinguish specific bird species.

To address this, we reviewed 1,231 label classes across the source datasets and curated a unified taxonomy with 178 label classes. These classes span a variety of Linnaean taxonomic ranks, with each class having a minimum of 100 annotations to ensure sufficient training examples. As with the video models, these label classes represent a narrow subset of the wildlife of interest to conservationists. This model is intended to primarily serve as a base model for fine-tuning custom models, rather than serving as a general-purpose model that directly covers conservation use cases.

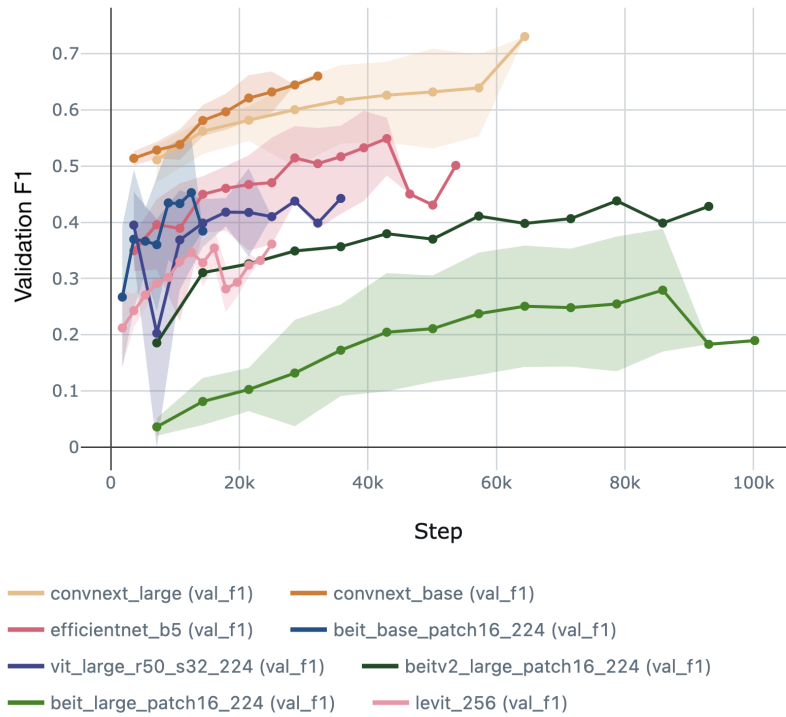
As with the video classification models, a split-aware strategy was used to create training, validation, and test sets. All data from a given camera location were assigned to the same

split to avoid overfitting to background features and ensure generalizable model performance.

### 5.1.3. Classification model architecture

In order to select a backbone architecture, we evaluated several modern neural network architectures that post-date the commonly used backbones in most existing camera trap models. The candidate architectures included ConvNeXt, BEiT, EfficientNet, LeViT, and ViT.

Figure 6 shows run curves for the different backbone architectures we evaluated. Runs for ConvNeXt variants outperformed others, achieving greater accuracy with fewer training epochs. Based on these results, we chose ConvNeXt V2 base as the backbone.



**Figure 6.** Validation F1 training run curves for candidate model architectures. The filled band shows the min and max range across runs, and the line shows the average across runs.

The chosen ConvNeXt V2 base model contains 87.7 million parameters and operates on  $224 \times 224$  pixel input crops. In Figure 6, its performance is represented by the dark orange curve.

## 5.2. Results

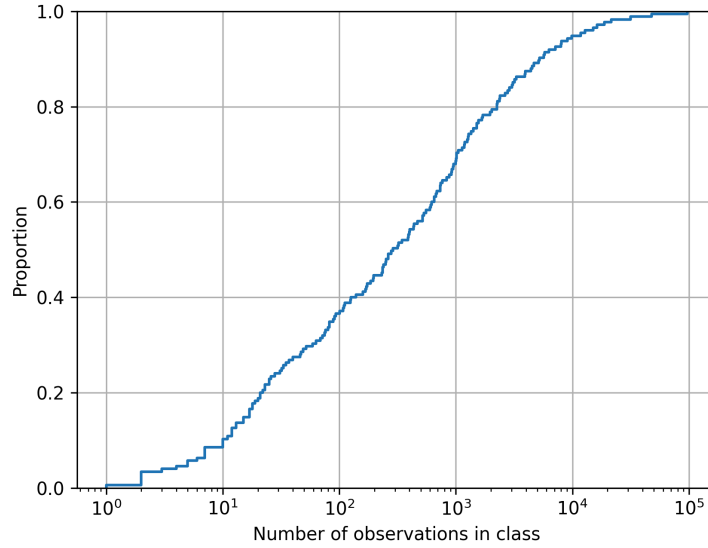
### 5.2.1. Pretrained model performance

As with videos, metrics for camera trap tasks are highly dataset-dependent and can vary widely based on deployment context and class balance. Here, we report some illustrative metrics for Zamba’s pretrained image model `lila.science` using holdout validation.

The holdout set reported here contains 449,263 randomly sampled observations held out from training. Splits were performed on a transect-by-transect basis<sup>12</sup> while ensuring at least one transect representing each label class is present in all splits. The holdout set is

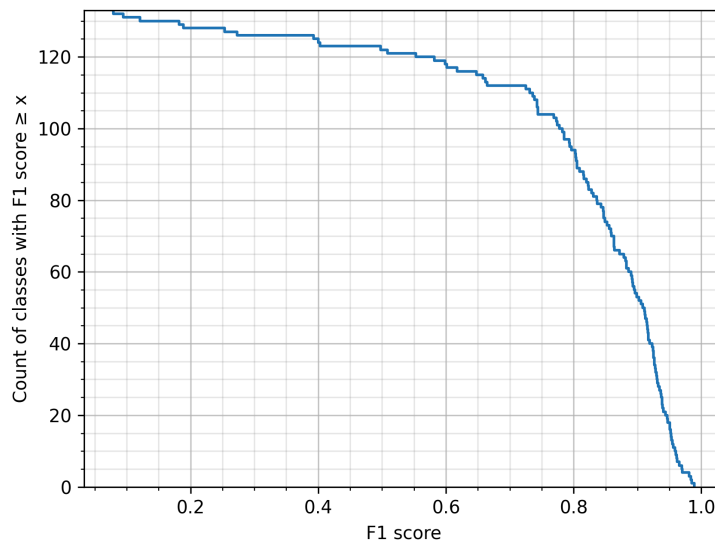
<sup>12</sup>This means all images from a given transect (camera location) are assigned entirely to one split. This methodology is the same as what we used for splitting the video data.

highly unbalanced, with an exponential-like distribution of class sizes. Figure 7 shows an empirical cumulative distribution function (ECDF) of the number of observations in each class.



**Figure 7.** ECDF of showing the distribution of class sizes. The curve shows the proportion of the label classes with up to the number of observations in the holdout set given by the x-axis value. Note that the x-axis is shown in log-scale.

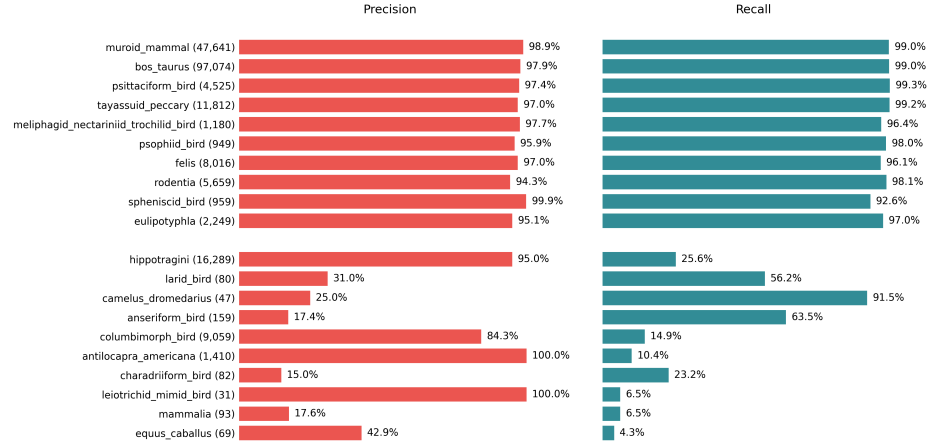
On the holdout set, the `lila.science` model achieved 90% top-1 accuracy. Performance varies widely by label class, although the majority of label classes see relatively strong performance. Figure 7 shows the distribution of F1 scores across label classes for classes with at least 30 observations. Of those (133 classes), 40% (53 classes) have an F1 score of over 0.9 and 71% (94) have an F1 score of over 0.8.



**Figure 8.** Complementary ECDF of F1 score for label classes in the holdout set with at least 30 observations. The curve shows the count of label classes whose F1 score exceeds the value given by the x-axis.



Figure 9 shows the precision and recall scores for top 10 and bottom 10 label classes when ranked by F1 score. This shows that the model performs highly accurately on the top performing label classes. For the weakest label classes, there is a mix of strong-precision-weak-recall, weak-precision-strong-recall, or weak performance in both metrics. The weakest classes tend to be classes with the fewest observations in the test set (<100), although some classes with many observations like hippotragini and columbimorph\_bird with thousands of test observations show relatively low recall scores. Cases of weak performance are often driven by confusion between phenotypically similar classes; for instance, observations involving types are birds are often predicted at different taxonomic granularity from the label.



**Figure 9.** Precision and recall values for a selection of label classes. The top half are the top 10 label classes ranked by F1 score, while the bottom half are the bottom 10 label classes ranked by F1 score. The parenthetical annotation gives the number of observations of that class.

For an indirect comparison<sup>13</sup>, SpeciesNet, the model used by Wildlife Insights, report a 81.9% accuracy, 80.7 species-weighted-average F1 score, and 54.2 macro-averaged F1 score on a test dataset of 42,791 images with 101 label classes[44]. This comparison shows that Zamba’s pretrained model is in an approximate comparable range of performance as the published results of similar tools.

## 6. DEPTH ESTIMATION FOR VIDEOS

Statistical models can be used to estimate animal abundance from camera trapping [59]. One such method is the distance sampling framework [14], which combines the frequency of animal sightings with the distance from the camera to each animal to estimate a species’ full population size [8]. In the following sections, we outline the training data, model architecture, and performance of Zamba’s depth estimation approach for videos.

### 6.1. Methods

As discussed in the “Development History” section above, the Deep Chimpact machine learning challenge was held to develop and train models for performing depth estimation on monocular camera trap videos.

<sup>13</sup>As discussed for video species classification, it is difficult to report meaningful direct comparisons to other tools or study results. There are not well-established standard benchmarking procedures and datasets, and the high degree of variance in attributes across camera trap deployments means that performance is highly sensitive to similarity to the training data, and that metrics reported on one dataset do not necessarily translate well to others.

The top performing model from the second place ensemble<sup>14</sup> was integrated into Zamba as an inference-only module.<sup>15</sup> As no additional retraining was conducted, the below sections describe the competition data and results.

### 6.1.1. Training data

Zamba's depth estimation model was developed using a unique dataset of approximately 3,900 videos, created by research teams from the Max Planck Institute for Evolutionary Anthropology (MPI-EVA) and the Wild Chimpanzee Foundation (WCF). Videos were from Taï National Park in Côte d'Ivoire and Moyen-Bafing National Park in the Republic of Guinea and captured six different species groups: bushbucks, chimpanzees, duikers, elephants, leopards, and monkeys.

Ground truth distances in the dataset were manually annotated with the aid of reference videos, which are recordings of field researchers walking away from each camera trap holding a sign up every meter to indicate how far they were from the camera. Such reference videos are a typical way to perform depth estimation for camera trap videos [14]; however, they are time and resource intensive to record and to use in labeling.



**Figure 10.** A researcher in Taï National Park, Côte d'Ivoire holding up a sign at 5 meters away as part of a depth reference video. Image courtesy of Wild Chimpanzee Foundation.

Videos in the original dataset that included more than one animal were excluded as there was not a way in the labeled data to identify which distance label corresponded to which animal. This removed 18% of frames from the original Taï National Park dataset and 22% of frames from the original Moyen-Bafing dataset.

Data was split evenly into a training dataset and a test dataset. Each site was either entirely in the train set or the test set, so all backgrounds used to evaluate submissions were new to the model.

**Table 5.** Counts per location for the final dataset used in the competition

Park	Train set	Test set
------	-----------	----------

<sup>14</sup>The second-place submission was chosen for modeling framework compatibility reasons. While the full competition-winning approach was an ensemble, the performance of the best single model was almost identical to the ensemble (MAE of 1.635m for the single model compared to 1.625m for the ensemble) so only one model was integrated into Zamba. This supports lower computing costs and faster inference speeds without sacrificing performance.

<sup>15</sup>Training and fine-tuning is not supported as of version v2.6.1 (May 2025).

Taï National Park, Côte d'Ivoire	530 videos 4,173 frames	508 videos 3,802 frames
Moyen-Bafing National Park, Guinea	1,542 videos 11,056 frames	1,328 videos 8,130 frames

**Table 6.** *Distribution of species labels in the train and test sets*

Species	Train videos	Train frames	Test videos	Test frames
duiker	744	6,805	695	5,718
buckbuck	261	3,550	230	2,603
chimpanzee	304	2,876	249	2,150
monkey	738	1,757	642	1,279
leopard	27	230	20	162
elephant	1	11	1	20

The videos ranged between 5 seconds and 1 minute long. Each annotation includes a timestamp in seconds since the start of the video and the ground truth distance in meters.

**Table 7.** *Example rows of depth estimation annotations*

video_id	time	distance
zxsx.mp4	0	7.0
zxsx.mp4	2	8.0
zxsx.mp4	4	10.0

### 6.1.2. No references videos

The depth estimation model was not trained using any reference videos directly. The model was instead required to learn purely from monocular images and the distance ground truth labels. This is a more challenging task but enables more complete automation of the workflow.

### 6.1.3. Model architecture

The depth estimation pipeline applies the following steps. First, videos are resampled to one frame per second to match the training dataset. Next, an object detection model is used on each frame to find frames with animals in them. The object detection model used is [MegadetectorLite](#). Then, the depth estimation model estimates distance between the animal and the camera for each frame.<sup>16</sup>

**Figure 11.** *Flow diagram of the video depth estimation pipeline. Gray nodes indicate data and blue nodes indicate a processing step.*

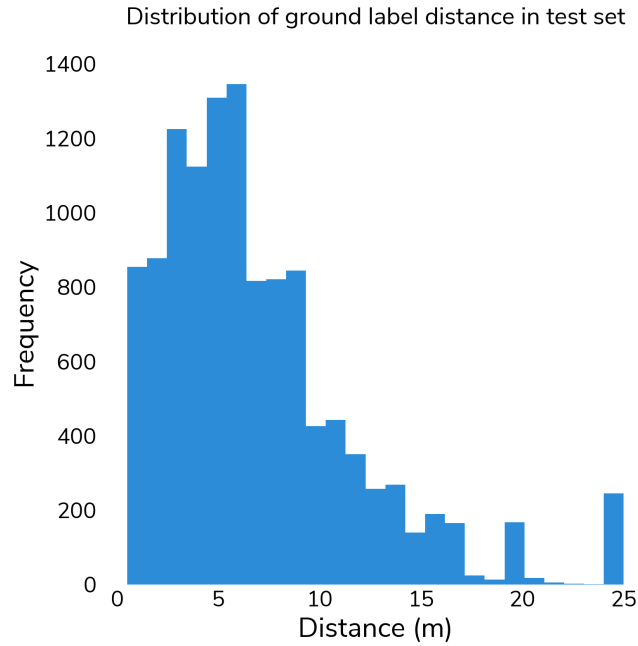
The depth estimation model uses an EfficientNetV2 [38] backbone and the output is a scalar estimated distance in meters between the animal and the camera. The inputs to the model are 5 frames stacked channelwise, where the frames are chronological and the middle frame corresponds to the present frame for which the depth estimation is made. The frames are downsampled to 270x480. The model was trained with heavy augmentations for 80

<sup>16</sup>Although the model was trained on single-individual videos, it produces an output for each detected animal in a frame. However, for frames with multiple individuals, the distance for all animals will be the same. If there is no animal in the frame, the distance will be null.

epochs with a batch size of 32, AdamW optimizer, and CosineAnnealingLR scheduler with a period of 25 epochs.<sup>17</sup>

## 6.2. Results

Model performance is reported for the test set of nearly 12,000 frames described above. The distribution of distances is shown in Figure 12. Nearly half (45%) of frames had a labeled distance of 5 meters or less and 80% of frames had a labeled distance of 10 meters or less.

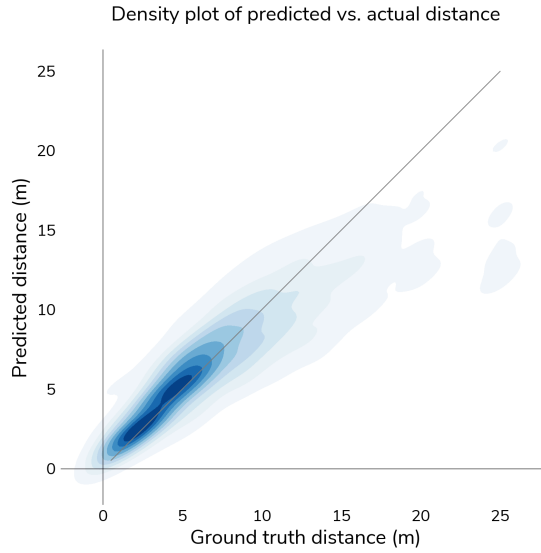


**Figure 12.** *Distribution of distances in the depth estimation test set*

The depth estimation model has a mean absolute error (MAE) of 1.635 m. The model is more accurate when predicting depth for animals closer to the camera. The MAE for animals  $\leq 10$  m from the camera trap is 1.06 m, while at further distances, the model underestimates distance. However, accuracy at closer distances is more important for distance sampling methods [60].

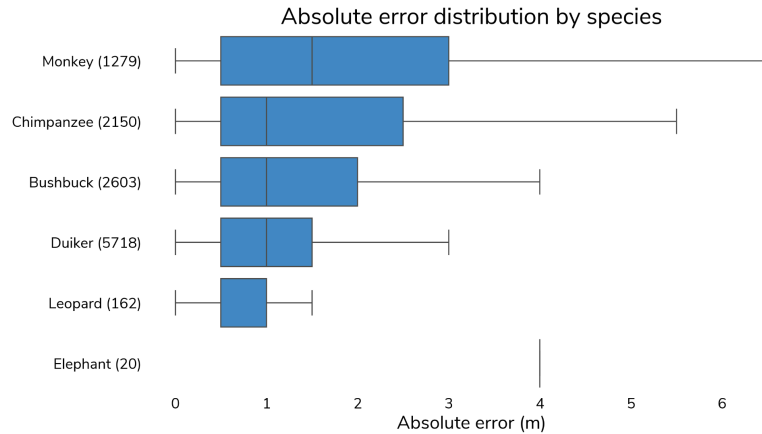
T. Haucke, H. S. Kühl, J. Hoyer, and V. Steinhage [14] applied machine learning to monocular depth *using reference videos* and reported a mean absolute error (MAE) of 1.85 m. The relatively similar performance (albeit on different datasets) suggests that accurate machine learning predictions are possible without reference videos.

<sup>17</sup>These hyperparameters were part of the second-place winning solution of the Deep Chimpact Challenge and found empirically.



**Figure 13.** *Density plot of predicted vs. actual distance for the depth estimation model*

The median absolute error was relatively similar across species, though the distribution of errors differs a bit. Figure 14 show the interquartile range (shaded portion) of absolute errors by species. Outliers extending beyond 1.5 times the interquartile range are not plotted.



**Figure 14.** *Distribution of test set absolute error by species.*

## 7. ZAMBA CLOUD

As a command-line program and Python library distributed as a package, Zamba is capable and easily deployable. However, there are still two major limitations. First, not all researchers have such hardware available with graphical processing units (GPUs) to run deep learning models in a practical way. Without this specialized hardware running Zamba's models on large datasets of videos or images is inefficient. Secondly, many wildlife conservationists are not programmers and may have little experience with command-line programs or scripting in Python, and it can be a barrier to create the environment and install the dependencies to run these tools.

In order to make Zamba easy to use and accessible to any camera trap researcher, a point-and-click graphical-user interface program was developed. **Zamba Cloud** is a web application with no-code workflows to use Zamba’s capabilities. Zamba Cloud currently has workflows that support the species classification for videos and species classification for images tasks. For both tasks, users can fine-tune from any of Zamba’s pretrained models, and they can perform inference on unlabeled videos/images using a pretrained model, one of their fine-tuned models, or a fine-tuned model that another user has chosen to make publicly available.

All of the workflows start with uploading videos or images using a drag-and-drop interface. Alternatively, uploads via an FTP server are also supported. One key tradeoff of implementing a web application, as opposed to a desktop application, is that users are required to have internet access with sufficient bandwidth. To mitigate the bandwidth requirements, Zamba Cloud implements an optional upload process that performs client-side video resizing. This option uses `ffmpeg.wasm` [61], a WebAssembly port of FFmpeg [62] to reduce the resolution and framerate of submitted videos to fixed sizes. If fine-tuning, users additionally upload label data and confirm whether they match to existing label classes or are new classes.



**Figure 15.** Screenshots showing the custom model fine-tuning workflow in Zamba Cloud.

Zamba Cloud runs all machine learning workloads on managed cloud infrastructure with GPUs. This means users can easily upload their data and use Zamba without needing specialized hardware or to install complex software.

For the inference workflow, Zamba Cloud has a review interface that allows users to easily visually inspect each image or video alongside predicted species classifications with confidence scores. For images, the bounding boxes from object detection are also shown. See [Figure 16](#) for an example screenshot.

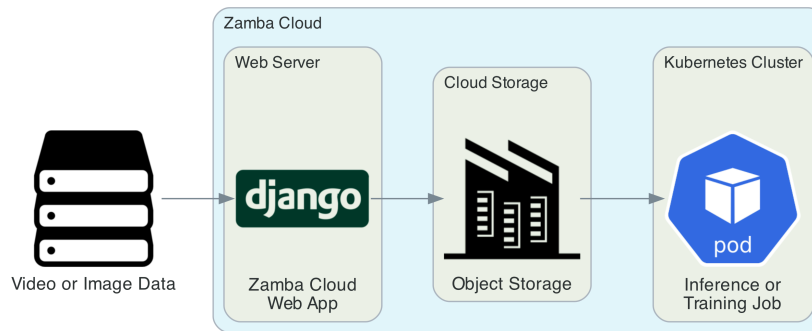


**Figure 16.** Screenshot of the prediction review interface in Zamba Cloud.

Predictions from the inference workflow are exportable in comma-separated values (CSV) format. CSV is chosen as a simple, open format that is portable and supported by common spreadsheet software.



Zamba Cloud's server is implemented in Python using Django [63]. Machine learning workloads are containerized and submitted as batch jobs to a Kubernetes cluster.



**Figure 17.** Architecture diagram of Zamba Cloud.

## 8. DISCUSSION

Zamba implements workflows for inference on unlabeled videos or images using pre-trained models as well as creating custom models by fine-tuning pretrained models with labeled data. Each of these tasks are available through the CLI or the Python library.

Custom model training represents Zamba's flagship contribution and unique capability among camera trap analysis tools. Camera traps are used across hugely variable habitats, taxonomic rank, and camera positions. A single general pretrained model could not accommodate fish detected by underwater deployments, large mammals in the savannah, and insects captured in a hole in the ground. Zamba's customization features dramatically expand applicability beyond users studying the species that existing pretrained models were trained on.

In addition, no-code workflows to use Zamba's capabilities on Zamba Cloud means custom machine learning models are now accessible to non-programmers. To the best of our knowledge, Zamba Cloud is currently the only tool that supports fine-tuning custom video-native machine learning algorithms for camera traps without writing any code. To date, over 300 users from around the globe have used Zamba Cloud to process more than 1.1 million videos.

### 8.1. Guidance on use

Zamba is designed to accelerate human workflows rather than replace human expertise, reducing the volume of videos that require manual review and removing procedural steps like identifying blank videos. Camera trap research often involves human coding (e.g., assessments of animal behavior) regardless of automated filtering or selection. At its current capability level, we recommend Zamba be used to inform research workflows rather than automate final outputs. The most common applications are:

- **Blank filtering:** Zamba can be used to [filter blank camera trap videos](#) using probabilistic classifications, saving researchers significant time and storage costs. Researchers can adjust [probability thresholds](#) based on their priorities—setting conservative thresholds to avoid losing any animal videos, or more aggressive ones to maximize blank video removal.
- **Targeted species search:** Zamba can be used to identify videos containing specific species of interest by sorting videos in descending order of probability for the target



species. [Targeted search](#) saves researchers significant time and allows them to get more value out of camera trap data. Even with an imperfect model, researchers can focus on high confidence predictions overall or high confidence predictions within specific labels. Targeted search also enables collaboration between research groups, as one team's bycatch can become another team's primary data source.

- **Custom models:** Custom models trained on user-provided data support use in new habitats or with different species. They can also be used to train site-specific models, which can yield more accurate predictions even if the species labels overlap with existing models, or [improve classification of rare species](#). We recommend users first evaluate existing pretrained models before investing in custom training. When developing custom models, researchers should understand that the amount of data needed depends greatly on the specific data and use case, so we recommend an iterative approach. Label distribution in the training data will also affect model behavior: rare classes are harder for models to learn effectively, while balanced datasets may produce more false positives for genuinely rare species in deployment.

## 9. CONCLUSION

Zamba is a powerful tool in supporting wildlife conservation. Initially developed to tackle the technically demanding task of processing camera-trap videos, it has since expanded to handle images as well. A key capability of Zamba is the ability to fine-tune models to better target specific habitats or expand to new subjects of interest, with support for species classification for both videos and images. The Zamba open source package supports programmatic use at the command line or as a Python library, while Zamba Cloud provides a no-code option for non-programmer users. The use of automated machine learning workflows for camera trap data can save countless hours of valuable time, speed up conservation interventions, and enable camera traps to be used to their fullest potential.

## ACKNOWLEDGMENTS

This project has been supported by funding from the Max Planck Institute for Evolutionary Anthropology, the Arcus Foundation, the Patrick J. McGovern Foundation, and the WILD-LABS Awards 2024.

The authors express their gratitude to collaborators, advisors, and data providers at the following organizations: the Max Planck Institute for Evolutionary Anthropology, the German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, the Helversen'sche Stiftung, and Dan Morris.

The authors thank the following PanAf collaborators for supporting and collecting the PanAf video dataset: Abel Nzeheke, Alexander Tickle, Amelia Meier, Anne-Celine Granjon, Anthony Agbor, Dervla Dowd, Emmanuel Ayuk Ayimisin, Emmanuel Dilambaka, Emmanuelle Normand, Fiona Stewart, Geoffrey Muhanguzi, Giovanna Maretti, Henk Eshuis, Hilde Vanleeuwe, Jean Claude Dengui, John Hart, Joshua M. Linder, Jospehine Head, Juan Lapuente, Karsten Dierks, Katherine Corogenes, Kathryn J. Jeffery, Kevin Lee, Lucy Jayne Ormsby, Manasseh Eno-Nku, Martijn Ter Heegde, Mohamed Kambi, Nadege Wangue Njomen, Parag Kadam, Paul Telfer, Robinson Orume, Samuel Angedakin, Sergio Marrocoli, Sorrel Jones, Theophile Desarmieux, Thurston Cleveland Hicks, Vera Leinert, Vianet Mihindou, Vincent Lapeyre, and Virginie Vergnes. The authors also thank the organizations and their members that have collaborated with the PanAf to collect this data: Budongo Conservation Field Station (Uganda), Fongoli Savanna Chimpanzee Project (Senegal), Gashaka Primate Project (Nigeria), Goualougo Triangle Ape Project, Greater Mahale Ecosystem Research and Conservation, Jane Goodall Institute Spain (Dindefelo, Senegal), Korup Rainforest Conservation Society (Cameroon), Lukuru Wildlife Research Foundation

(DRC), Ngogo Chimpanzee Project (Uganda), Ozouga Chimpanzee Project and Loango Gorilla Project (Gabon), Station d'Etudes des Gorilles et Chimpanzees (Gabon), Tai Chimpanzee Project (Cote d'Ivoire), the Aspinall Foundation (Gabon), WCS (Conkouati-Douli NP, R-Congo), Wild Chimpanzee Foundation (Cote d'Ivoire, Guinea, Liberia), Wildlife Conservation Society (WCS Nigeria), WWF (Campo Ma'an NP, Cameroon), and WWF Congo Basin (DRC). The authors further thank the PanAf project and data managers: Paula Dieguez, Mizuki Murai, and Yasmin Moebius. The authors also thank the PanAf video data cleaners and Chimp&See community scientists and moderators who annotated the PanAf video data: Nuria Maldonado, Anja Landsmann, Laura K. Lynn, Zuzana Rockaiova, Heidi Pfund, Heike Wilken, Lucia Hacker, Libby Smith, Karen Harvey, Tonnie Cummings, Carol Elkins, Briana Harder, Kristeena Sigler, Jane Widness, Amelie Pettrich, Antonio Buzharevski, Eva Martinez Garcia, Ivana Kirchmair, Sebastian Schütte, Joana Pereira, Silke Atmaca, Sadie Tenpas, and all [community scientists](#). The authors extend their deepest gratitude to the government organizations that have permitted data collection in their countries: Ministère de la Recherche Scientifique et de l'Innovation, Cameroon; Ministère des Forêts et de la Faune, Cameroon; Ministère des Eaux et Forêts, Cote d'Ivoire; Ministère de l'Enseignement Supérieur et de la Recherche Scientifique, Côte d'Ivoire; Agence Nationale des Parcs Nationaux, Gabon; Centre National de la Recherche Scientifique, Gabon; Ministère de l'Agriculture de l'Elevage et des Eaux et Forêts, Guinea; Forestry Development Authority, Liberia; National Park Service, Nigeria; Ministère de l'Economie Forestière, R-Congo; Ministère de la Recherche Scientifique et Technologique, R-Congo; Direction des Eaux, Forêts et Chasses, Senegal; Tanzania Commission for Science and Technology, Tanzania; Tanzania Wildlife Research Institute, Tanzania; Makerere University Biological Field Station Uganda; Uganda National Council for Science and Technology, Uganda; Uganda Wildlife Authority, Uganda; National Forestry Authority, Uganda. The authors also thank the generous funders of PanAf: Max Planck Society, Max Planck Society Innovation Fund, Heinz L. Krekeler Foundation, Arcus Foundation, the Patrick J. McGovern Foundation, Google AI for Social Good, and Facebook.

The authors thank the other contributors to the Zamba and Zamba Cloud codebases: Justin Chung Clark, Casey Fitzpatrick, Robert Gibboni, Tamara Glazer, Isaac Slavitt, Stan Triepels, and Katie Wetstone. The authors also thank Dmytro Poplavskiy, the winner of the Pri-Matrix Challenge, whose winning solution was the basis for the original Zamba V1 video species classification model; and Kirill Brodt, the second-place winner of the Deep Chimpact Challenge, whose solution is the basis for the depth estimation model. The authors thank Hannah Moshontz de la Rocha for review and edits to the manuscript. Finally, the authors thank the reviewers who provided constructive comments and suggestions to improve the quality of the manuscript.

## REFERENCES

- [1] A. C. Burton et al., "REVIEW: Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes," *Journal of Applied Ecology*, vol. 52, no. 3, pp. 675–685, 2015, doi: [10.1111/1365-2664.12432](https://doi.org/10.1111/1365-2664.12432).
- [2] M. Arandjelovic et al., "Highly precise community science annotations of video cameratrapped fauna in challenging environments," *Remote Sensing in Ecology and Conservation*, vol. 10, no. 6, pp. 702–724, 2024, doi: [10.1002/rse2.402](https://doi.org/10.1002/rse2.402).
- [3] J. Véléz et al., "An evaluation of platforms for processing cameratrap data using artificial intelligence," *Methods in Ecology and Evolution*, vol. 14, no. 2, pp. 459–477, 2022, doi: [10.1111/2041-210X.14044](https://doi.org/10.1111/2041-210X.14044).
- [4] DrivenData, P. Bull, E. Dorne, R. Gibboni, J. Qi, and K. Wetstone, "Zamba." [Online]. Available: <https://zenodo.org/doi/10.5281/zenodo.15116959>
- [5] F. Rovero and R. Kays, "Camera trapping for conservation," in *Conservation Technology*, Oxford University Press/Oxford, 2021, pp. 79–104. doi: [10.1093/oso/9780198850243.003.0005](https://doi.org/10.1093/oso/9780198850243.003.0005).
- [6] M. W. Tobler, A. Zúñiga Hartley, S. E. CarrilloPercastegui, and G. V. N. Powell, "Spatiotemporal hierarchical modelling of species richness and occupancy using camera trap data," *Journal of Applied Ecology*, vol. 52, no. 2, pp. 413–421, 2015, doi: [10.1111/1365-2664.12399](https://doi.org/10.1111/1365-2664.12399).

- [7] R. Sollmann, A. Mohamed, H. Samejima, and A. Wilting, "Risky business or simple solution – Relative abundance indices from camera-trapping," *Biological Conservation*, vol. 159, pp. 405–412, 2013, doi: [10.1016/j.biocon.2012.12.025](https://doi.org/10.1016/j.biocon.2012.12.025).
- [8] E. J. Howe, S. T. Buckland, M. DesprésEinspenner, and H. S. Kühl, "Distance sampling with camera traps," *Methods in Ecology and Evolution*, vol. 8, no. 11, pp. 1558–1565, 2017, doi: [10.1111/2041-210x.12790](https://doi.org/10.1111/2041-210x.12790).
- [9] T. G. O'Brien et al., "Camera trapping reveals trends in forest duiker populations in African National Parks," *Remote Sensing in Ecology and Conservation*, vol. 6, no. 2, pp. 168–180, 2019, doi: [10.1002/rse2.132](https://doi.org/10.1002/rse2.132).
- [10] A. Caravaggi et al., "A review of camera trapping for conservation behaviour research," *Remote Sensing in Ecology and Conservation*, vol. 3, no. 3, pp. 109–122, 2017, doi: [10.1002/rse2.48](https://doi.org/10.1002/rse2.48).
- [11] M. Bessone et al., "Drawn out of the shadows: Surveying secretive forest species with camera trap distance sampling," *Journal of Applied Ecology*, vol. 57, no. 5, pp. 963–974, 2020, doi: [10.1111/1365-2664.13602](https://doi.org/10.1111/1365-2664.13602).
- [12] S. E. Green, J. P. Rees, P. A. Stephens, R. A. Hill, and A. J. Giordano, "Innovations in Camera Trapping Technology and Approaches: The Integration of Citizen Science and Artificial Intelligence," *Animals*, vol. 10, no. 1, p. 132, 2020, doi: [10.3390/ani10010132](https://doi.org/10.3390/ani10010132).
- [13] S. Beery, D. Morris, and S. Yang, "Efficient Pipeline for Camera Trap Image Review." [Online]. Available: <https://arxiv.org/abs/1907.06772>
- [14] T. Haucke, H. S. Kühl, J. Hoyer, and V. Steinhage, "Overcoming the distance estimation bottleneck in estimating animal abundance with camera traps," *Ecological Informatics*, vol. 68, p. 101536, 2022, doi: [10.1016/j.ecoinf.2021.101536](https://doi.org/10.1016/j.ecoinf.2021.101536).
- [15] J. A. Ahumada et al., "Wildlife Insights: A Platform to Maximize the Potential of Camera Trap and Other Passive Sensor Wildlife Data for the Planet," *Environmental Conservation*, vol. 47, no. 1, pp. 1–6, 2019, doi: [10.1017/s0376892919000298](https://doi.org/10.1017/s0376892919000298).
- [16] P. van Lunteren, "AddaxAI: A no-code platform to train and deploy custom YOLOv5 object detection models," *Journal of Open Source Software*, vol. 8, no. 88, p. 5581, 2023, doi: [10.21105/joss.05581](https://doi.org/10.21105/joss.05581).
- [17] P. Bull, I. Slavitt, and G. Lipstein, "Harnessing the Power of the Crowd to Increase Capacity for Data Science in the Social Sector." [Online]. Available: <https://arxiv.org/abs/1606.07781>
- [18] DrivenData, "Pri-matrix Factorization." [Online]. Available: <https://www.drivendata.org/competitions/49/deep-learning-camera-trap-animals/>
- [19] DrivenData, "Results from Pri-Matrix Factorization and a New Open Source Tool for Wildlife Research and Conservation." [Online]. Available: <https://drivendata.co/blog/camera-trap-wildlife-winners>
- [20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." [Online]. Available: <https://arxiv.org/abs/1512.03385>
- [21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision." [Online]. Available: <https://arxiv.org/abs/1512.00567>
- [22] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions." [Online]. Available: <https://arxiv.org/abs/1610.02357>
- [23] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning." [Online]. Available: <https://arxiv.org/abs/1602.07261>
- [24] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in KDD '16. 2016, pp. 785–794. doi: [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785).
- [25] DrivenData, "Deep Chimpanzee: Depth Estimation for Wildlife Conservation." [Online]. Available: <https://www.drivendata.org/competitions/82/competition-wildlife-video-depth-estimation/>
- [26] A. K. Kalan et al., "Novelty Response of Wild African Apes to Camera Traps," *Current Biology*, vol. 29, no. 7, pp. 1211–1217, 2019, doi: [10.1016/j.cub.2019.02.024](https://doi.org/10.1016/j.cub.2019.02.024).
- [27] M. S. McCarthy et al., "Camera traps provide a robust alternative to direct observations for constructing social networks of wild chimpanzees," *Animal Behaviour*, vol. 157, pp. 227–238, 2019, doi: [10.1016/j.anbehav.2019.08.008](https://doi.org/10.1016/j.anbehav.2019.08.008).
- [28] C. Boesch et al., "Chimpanzee ethnography reveals unexpected cultural diversity," *Nature Human Behaviour*, vol. 4, no. 9, pp. 910–916, 2020, doi: [10.1038/s41562-020-0890-1](https://doi.org/10.1038/s41562-020-0890-1).
- [29] B. Debetencourt, M. M. Barry, M. Arandjelovic, C. Stephens, N. Maldonado, and C. Boesch, "Camera traps unveil demography, social structure, and home range of six unhabituated Western chimpanzee groups in the Moyen Bafing National Park, Guinea," *American Journal of Primatology*, vol. 86, no. 2, 2023, doi: [10.1002/ajp.23578](https://doi.org/10.1002/ajp.23578).
- [30] M. S. McCarthy et al., "Chimpanzee identification and social network construction through an online citizen science platform," *Ecology and Evolution*, vol. 11, no. 4, pp. 1598–1608, 2020, doi: [10.1002/ece3.7128](https://doi.org/10.1002/ece3.7128).
- [31] J. S. Head, C. Boesch, M. M. Robbins, L. I. Rabanal, L. Makaga, and H. S. Kühl, "Effective sociodemographic population assessment of elusive species in ecology and conservation management," *Ecology and Evolution*, vol. 3, no. 9, pp. 2903–2916, 2013, doi: [10.1002/ece3.670](https://doi.org/10.1002/ece3.670).

- [32] M. Henrich et al., “A semiautomated camera trap distance sampling approach for population density estimation,” *Remote Sensing in Ecology and Conservation*, vol. 10, no. 2, pp. 156–171, 2023, doi: [10.1002/rse2.362](https://doi.org/10.1002/rse2.362).
- [33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi: [10.1109/cvpr.2016.90](https://doi.org/10.1109/cvpr.2016.90).
- [34] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A Closer Look at Spatiotemporal Convolutions for Action Recognition,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. doi: [10.1109/cvpr.2018.00675](https://doi.org/10.1109/cvpr.2018.00675).
- [35] G. Bertasius, H. Wang, and L. Torresani, “Is Space-Time Attention All You Need for Video Understanding?,” [Online]. Available: <https://arxiv.org/abs/2102.05095>
- [36] C. Feichtenhofer, “X3D: Expanding Architectures for Efficient Video Recognition,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 200–210. doi: [10.1109/cvpr42600.2020.00028](https://doi.org/10.1109/cvpr42600.2020.00028).
- [37] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. doi: [10.1109/cvpr.2017.502](https://doi.org/10.1109/cvpr.2017.502).
- [38] M. Tan and Q. V. Le, “EfficientNetV2: Smaller Models and Faster Training,” 2021, doi: [10.48550/ARXIV.2104.00298](https://doi.org/10.48550/ARXIV.2104.00298).
- [39] fastai developers, “TimeDistributed.” [Online]. Available: <https://docs.fast.ai/layers.html#timedistributed>
- [40] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “SlowFast Networks for Video Recognition.” [Online]. Available: <https://arxiv.org/abs/1812.03982>
- [41] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO Series in 2021.” [Online]. Available: <https://arxiv.org/abs/2107.08430>
- [42] G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network.” [Online]. Available: <https://arxiv.org/abs/1503.02531>
- [43] DrivenData, “What’s a takahē?,” [Online]. Available: <https://drivendata.co/blog/zamba-find-takahe>
- [44] T. Gadot et al., “To crop or not to crop: Comparing wholeimage and cropped classification on a large dataset of camera trap images,” *IET Computer Vision*, vol. 18, no. 8, pp. 1193–1208, 2024, doi: [10.1049/cvi2.12318](https://doi.org/10.1049/cvi2.12318).
- [45] D. Morris, “Labeled Information Library of Alexandria: Biology and Conservation.” [Online]. Available: <https://lila.science/>
- [46] S. Beery, G. Van Horn, and P. Perona, “Recognition in Terra Incognita,” in *Computer Vision – ECCV 2018*, Springer International Publishing, 2018, pp. 472–489. doi: [10.1007/978-3-030-01270-0\\_28](https://doi.org/10.1007/978-3-030-01270-0_28).
- [47] The Nature Conservancy, “Channel Islands Camera Traps.” [Online]. Available: <https://lila.science/datasets/channel-islands-camera-traps/>
- [48] Desert Lion Conservation Project, “Desert Lion Conservation Camera Traps.” [Online]. Available: <https://lila.science/datasets/desert-lion-conservation-camera-traps/>
- [49] H. Yousif, R. Kays, and Z. He, “Dynamic Programming Selection of Object Proposals for Sequence-Level Animal Species Classification in the Wild,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
- [50] Idaho Department of Fish and Game, “Idaho Camera Traps.” [Online]. Available: <https://lila.science/datasets/idaho-camera-traps/>
- [51] Island Conservation, “Island Conservation Camera Traps.” [Online]. Available: <https://lila.science/datasets/island-conservation-camera-traps/>
- [52] Z. Zhang, Z. He, G. Cao, and W. Cao, “Animal Detection From Highly Cluttered Natural Scenes Using Spatiotemporal Object Region Proposals and Patch Verification,” *IEEE Transactions on Multimedia*, vol. 18, no. 10, pp. 2079–2092, 2016, doi: [10.1109/tmm.2016.2594138](https://doi.org/10.1109/tmm.2016.2594138).
- [53] M. A. Tabak et al., “Machine learning to classify animal species in camera trap images: Applications in ecology,” *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 585–590, 2018, doi: [10.1111/2041-210x.13120](https://doi.org/10.1111/2041-210x.13120).
- [54] New Zealand Trailcams, “Trail Camera Images of New Zealand Animals.” [Online]. Available: <https://lila.science/datasets/nz-trailcams>
- [55] L. E. Pardo et al., “Snapshot Safari: A large-scale collaborative to monitor Africa’s remarkable biodiversity,” *South African Journal of Science*, vol. 117, no. 1/2, 2021, doi: [10.17159/sajs.2021/8134](https://doi.org/10.17159/sajs.2021/8134).
- [56] Saola Working Group, “Northern and Central Annamites Camera Traps 2.0.” [Online]. Available: <https://lila.science/datasets/swg-camera-traps>
- [57] Wildlife Conservation Society, “WCS Camera Traps.” [Online]. Available: <https://lila.science/datasets/wcscameratraps>
- [58] V. Anton, S. Hartley, A. Geldenhuis, and H. U. Wittmer, “Monitoring the mammalian fauna of urban areas using remote cameras and citizen science,” *Journal of Urban Ecology*, vol. 4, no. 1, 2018, doi: [10.1093/jue/juy002](https://doi.org/10.1093/jue/juy002).
- [59] N. A. Gilbert, J. D. J. Clare, J. L. Stenglein, and B. Zuckerberg, “Abundance estimation of unmarked animals based on cameratrap data,” *Conservation Biology*, vol. 35, no. 1, pp. 88–100, 2020, doi: [10.1111/cobi.13517](https://doi.org/10.1111/cobi.13517).

- [60] S. T. Buckland, D. R. Anderson, K. P. Burnham, J. L. Laake, D. L. Borchers, and L. Thomas, *Introduction to Distance Sampling: Estimating Abundance of Biological Populations*. Oxford University PressOxford, 2001. doi: [10.1093/oso/9780198506492.001.0001](https://doi.org/10.1093/oso/9780198506492.001.0001).
- [61] ffmpeg.wasm developers, “ffmpeg.wasm.” [Online]. Available: <https://github.com/ffmpegwasm/ffmpeg.wasm>
- [62] FFmpeg developers, “FFmpeg.” [Online]. Available: <https://ffmpeg.org/>
- [63] Django developers, “Django.” [Online]. Available: <https://www.djangoproject.com/>