applied sciences

MDPI

Article

Action Detection for Wildlife Monitoring with Camera Traps Based on Segmentation with Filtering of Tracklets (SWIFT) and Mask-Guided Action Recognition (MAROON)

Frank Schindler [1,*], Volker Steinhage [1], Suzanne T. S. van Beeck Calkoen [2,3,4] and Marco Heurich [2,5,6]

[1] Department of Computer Science IV, University of Bonn, Friedrich-Hirzebruch-Allee 8, 53115 Bonn, Germany; steinhag@cs.uni-bonn.de
[2] Department of National Park Monitoring and Animal Management, Bavarian Forest National Park, Freyunger Str. 2, 94481 Grafenau, Germany; suzanne.van_beeck_calkoen@tu-dresden.de (S.v.B.C.); marco.heurich@npv-bw.bayern.de (M.H.)
[3] Forest Zoology, Institute of Forest Botany and Forest Zoology, Technical University of Dresden, Pienner Str. 7, 01737 Tharandt, Germany
[4] Wildlife Sciences, Faculty of Forest Sciences and Forest Ecology, University of Göttingen, Büsgenweg 3, 37077 Göttingen, Germany
[5] Faculty of Environment and Natural Resources, University of Freiburg, Tennenbacher Straße 4, 79106 Freiburg im Breisgau, Germany
[6] Institute of Forestry and Wildlife Management, Inland Norway University of Applied Science, NO-2480 Koppang, Norway
* Correspondence: schindler@cs.uni-bonn.de

Abstract: Behavioral analysis of animals in the wild plays an important role for ecological research and conservation and has been mostly performed by researchers. We introduce an action detection approach that automates this process by detecting animals and performing action recognition on the detected animals in camera trap videos. Our action detection approach is based on SWIFT (segmentation with filtering of tracklets), which we have already shown to successfully detect and track animals in wildlife videos, and MAROON (mask-guided action recognition), an action recognition network that we are introducing here. The basic ideas of MAROON are the exploitation of the instance masks detected by SWIFT and a triple-stream network. The instance masks enable more accurate action recognition, especially if multiple animals appear in a video at the same time. The triple-stream approach extracts features for the motion and appearance of the animal. We evaluate the quality of our action recognition on two self-generated datasets, from an animal enclosure and from the wild. These datasets contain videos of red deer, fallow deer and roe deer, recorded both during the day and night. MAROON improves the action recognition accuracy compared to other state-of-the-art approaches by an average of 10 percentage points on all analyzed datasets and achieves an accuracy of 69.16% on the Rolandseck Daylight dataset, in which 11 different action classes occur. Our action detection system makes it possible todrasticallyreduce the manual work of ecologists and at the same time gain new insights through standardized results.

Keywords: wildlife monitoring; deep learning; video instance segmentation; mask-supported action recognition; triple-stream convolutional neural network; action detection for deer

Citation: Schindler, F.; Steinhage, V.; van Beeck Calkoen, S.; Heurich, M. Action Detection for Wildlife Monitoring with Camera Traps Based on Segmentation with Filtering of Tracklets (SWIFT) and Mask-Guided Action Recognition (MAROON) . *Appl. Sci.* **2024**, *14*, 514. https://doi.org/10.3390/app14020514

Academic Editors: Manuel Jesús Rodríguez Valido, Fernando Perez Nava and Gustavo Sutter

Received: 29 November 2023
Revised: 3 January 2024
Accepted: 4 January 2024
Published: 6 January 2024

1. Introduction

Understanding animal behavior is a fundamental component of conservation biology [1]. Consequently, action detection for wild animals has become an essential task for ecologists to assist conservation efforts [2]. However, quantifying the behavior of wild animals presents significant challenges and is often neglected as a result of the tremendous work-load that is needed to analyze all collected data.

Camera traps have become an ubiquitous tool in ecology and conservation that offer a reliable, minimally invasive and visual means of surveying wildlife [2–4]. Over the last few

decades, camera traps have been adopted for various ecological tasks, including abundance estimation [5–7], the quantification of species diversity [8], the detection of rare species [9], the investigation of animal activity patterns [10], and the analysis of species replacement processes [11]. Automatic analysis using artificial intelligence is absolutely necessary to deal with the vast amount of collected camera trap data [12,13]. Currently, many AI models are created that enable the management and processing of camera trap images and videos, facilitate the categorization of camera trap images, or are able to classify species within these images [14]. Although a few AI models exist that allow the classification of species in camera trap videos, a tool for the automatic quantification of animal behavior is still lacking.

Action recognition or action classification describes the task of assigning an action label to a (short) video sequence [15,16]. In that video sequence, only one actor is performing one action. In contrast, action detection is the combined task of detecting an object in a video and then performing action recognition on the found object. Therefore, with action detection it is possible to describe different actions from different actors at the same time in a video. Sometimes, however, the term action detection is used not only in a spatial context but also in a temporal context, meaning that it is not the person or animal that is spatially detected in a video but the start and end point of an action within a video that is determined. In this case, (temporal) action detection again refers to a single individual in a video and not to multiple actors.

Action recognition and action detection for wildlife monitoring are still relatively unexplored research areas. The following approaches represent all the important work in the field of action recognition and action detection in wildlife monitoring. General action recognition and action detection approaches are presented in Section 2. In our previous work [17], we considered action recognition for different European animals with just three simple action categories (walking, standing and eating). The work of [18] creates a neural network for the action recognition of great apes based on a self-created dataset. The Animal Kingdom Dataset is introduced in [19] and comprises action clips of a huge variety of animal species. The authors also propose an action recognition network, CARe, that recognizes the actions of unseen animal species. In the paper [20], the authors introduce a metric learning system with a triple-stream embedding network using RGB and optical flow features for the action recognition of great apes.

This paper presents the first action detection approach for camera trap videos. We combine our successful instance segmentation system SWIFT (segmentation with filtering of tracklets; see Section 3 for more detail) [21] with our newly developed action recognition network MAROON (mask-guided action recognition). The two most important innovations in our action recognition network are (1) the use of masked input frames and (2) the triple-stream approach for the spatio-temporal feature extraction. We evaluate our action recognition network on three different datasets and compare it to the state-of-the-art networks SlowFast [22] and MViT [23].

2. Related Work

In recent years, many action recognition and action detection approaches have been developed, which can be found in the surveys [15,16,24–26].

Three-dimensional convolutional neural networks (3D CNNs) are a famous approach to extract spatio-temporal features from videos to classify actions [27–31]. They outperform 2D convolutional approaches for action recognition [32,33]. The so-called X3D approach [34] forms a network architecture between 3D and 2D CNNs by extending 2D convolutional networks in space, time, width and depth to extract spatio-temporal features for action recognition. The most successful 3D CNNs are two-stream approaches, which were first introduced by [35]. SlowFast [22] represents one of the most successful two-stream convolutional neural networks by extracting primarily spatial features in one stream and temporal features in the other. The approach of [36] introduces a three-stream network using different amounts of frames in each stream. The combined features are fed into an LSTM (long short-term memory) network for action prediction. Another three-stream

convolutional neural network (3SCNN) that performs action recognition in 3D using the skeleton information of the actors is presented by [37] .

Several approaches exist that introduce modules that extend upon existing CNNs and improve the ability to extract spatio-temporal features. Examples are the Temporal Relation Network (TRN) [32], SpatioTemporal Module (STM) [38], Temporal Shift Module (TSM) [39], Temporal Excitation and Aggregation (TEA) [40], MotionSqueeze [41] and TokenLearner [42]. In addition, the Contrastive Action Representation Learning (CARL) framework [43] specializes in learning long-term action representations from videos over a very long time.

Transformer architectures have also become very popular for performing action recognition [44–46]. A very successful variant is the Multiscale Vision Transformer (MViT) [23], which combines the transformer architecture with the hierarchical multiscale feature extraction.

Using instance masks to support action recognition has been a poorly explored area. A few approaches exist, but these are either specialized to one specific application area or are limited in other aspects. SegCodeNet [47] uses a Mask R-CNN [48] to generate instance masks and combines them with the input frames in a two-stream approach to recognize different activities of wearables. The work of [49] uses a Mask R-CNN to extract the instance masks of one single actor in a video sequence and mask the frames accordingly. The masked frames are then fed into a single-stream CNN for action recognition. The work of [50] uses RGB frames overlaid with human pose information in a two-stream network to predict actions in the context of table tennis.

Action detection approaches are usually a combination of a detector and an action recognition network. However, because these two tasks are typically evaluated and analyzed separately, there is little work that presents comprehensive designs for action detection. Moreover, the quality of the action recognition network heavily depends on the quality of the detector when both are evaluated as one approach. Nevertheless, there are a few relevant action detection approaches in the literature. The Asynchronous Interaction Aggregation network [51] focuses on interactions between persons. The work of [52] introduces a baseline approach for weakly supervised action detection with multiple actors. The Efficient Video Action Detector (EVAD) [53] uses a transformer architecture to localize actors and classify the actions. The previously mentioned SlowFast network [22] is also able to perform action detection. The authors integrate a detector similar to Faster R-CNN in their architecture for detecting the actors and then perform action recognition on the found actors. The approach of [54] combines a SlowFast network with a human detector for action detection of persons.

Our action detection approach uses SWIFT as the detector and tracker. The functionality of SWIFT in the application area of wildlife monitoring is shown in our publication [21]. Our newly introduced action recognition network MAROON builds on SlowFast [22]. Therefore, we include the SlowFast approach in our evaluations. Moreover, it is a successful representative of the convolutional neural network approaches. We also analyze MViT [23], because transformer architectures are the other main approach to action recognition and MViT is one of the most successful networks.

3. Materials and Methods

3.1. Datasets

Annotated datasets in the context of action detection for wildlife monitoring are rare. Moreover, to the best of our knowledge, datasets that are suitable for instance segmentation, tracking and action recognition do not yet exist. To evaluate our action recognition network MAROON, we use three different self-created and annotated wildlife datasets: the Rolandseck Daylight, Bavarian Forest Daylight and Bavarian Forest Nighttime datasets.

With permission of the Wildpark Rolandseck GmbH (Remagen-Rolandseck, Germany) we captured video footage of fallow deer *(Dama dama)* and red deer *(Cervus elaphus)* in their natural environment in the Wildpark Rolandseck (Germany) from November 2020 to December 2021, resulting in over 6000 recorded videos. These videos were first viewed,

and incorrect recordings (e.g., no animals are present, camera was tilted extremely) were sorted out. This resulted in approximately 3000 videos. Because the data annotation forms a bottleneck, we had to choose representative videos for different scenarios, backgrounds, animals present and actions. Within the 3000 videos, there are often scenes that show similar situations and behavior. Using similar videos is not helpful for training neural networks. We used two Victure HC500 Trail Cameras (Victure, Shenzhen, China) and placed them at varying sites to be able to test changing backgrounds and different settings. The general site locations stayed the same, but we changed the exact position of the cameras each months to prevent always showing the same background. The manual annotation was performed with the VGG Image Annotator (VIA) Version 2.0.8 [55] and the annotation tool of [56]. For each animal in a frame, a segmentation mask, a bounding box, a class label, a track ID and an action label were assigned. All videos are 30 s long with 30 fps (frames per second) and a high-definition resolution of 1728 × 1296 pixels. Some videos in the dataset were shortened if there were no animals visible in the video anymore. Our annotated Rolandseck Daylight dataset consists of 33 videos. These videos are all captured during daylight. The following 11 different actions are present in the videos: foraging moving, foraging standing, grooming, head lowering, head raising, resting, running, standing up, vigilant lying, vigilant standing, and walking. A definition of all action classes is given in Appendix A. With this dataset, we already evaluated the effectiveness of our instance segmentation and tracking system SWIFT in our publication [21].

The Bavarian Forest Daylight and Bavarian Forest Nighttime datasets were captured in the Bavarian Forest National Park. The videos show roe deer *(Capreolus capreolus)* and red deer *(Cervus elaphus)*. These videos were selected from a larger dataset based on the number of individuals and different types of behavior present. The daylight videos are recorded with 15 FPS and a resolution of 1280 × 720. The nighttime videos have the same resolution but only 8 FPS. As the names of the datasets suggest, they are separated into daylight and nighttime videos. In contrast to the Rolandseck Daylight dataset, these two datasets generally show a lower number of individuals per video. This represents an important difference that will be of interest when investigating the influence of instance masks for our system. The following 7 different action classes are represented in this dataset: foraging standing, grooming, head lowering, head raising, vigilant standing, walking and sudden rush.

In Table 1, we provide an overview of the three datasets used. In this paper, we evaluate the action recognition ability of MAROON with these three datasets. In Figure 1, we show a exemplary frame from each of the datasets.

Table 1. Overview of the three datasets used.

Dataset	No. of Videos	Frames per Second (FPS)	Resolution	Avg. No. of Animals per Video	Different Action Classes
Rolandseck Daylight	33	30	1728 × 1296	5.88	11
Bavarian Forest Daylight	54	15	1280 × 720	1.13	7
Bavarian Forest Nighttime	31	8	1280 × 720	2.15	7

Figure 1. From left to right, exemplary frames from Rolandseck Daylight, Bavarian Forest Daylight and Bavarian Forest Nighttime datasets.

Before it is possible to use the data to train and test models, some preprocessing of the data is necessary. The videos from the datasets can contain multiple animals, and the animals perform different actions throughout one single video. To be usable for action recognition, each sequence must contain exactly one animal that performs only one action. Therefore, we extract each animal individually from the videos using the given bounding box information. We then split this video of the individual animal at the time points where the animal's action changes. Thus, from the complete videos of the datasets, many (action) sequences are created, which show only one animal performing only one action during the whole sequence. The length of the sequences can vary from a short section (1 s) to a total video length (30 s). It depends only on how long an action is performed. In Figure 2, we show the distribution of the action class sequences in our three datasets. As is common in the field of action recognition, especially in the field of wildlife monitoring, long-tailed distributions are present in all datasets [57]. This means that the action classes are not evenly distributed in the dataset. For example, the classes walking, vigilant standing, and foraging standing are more common than the classes grooming, resting, or sudden rush.

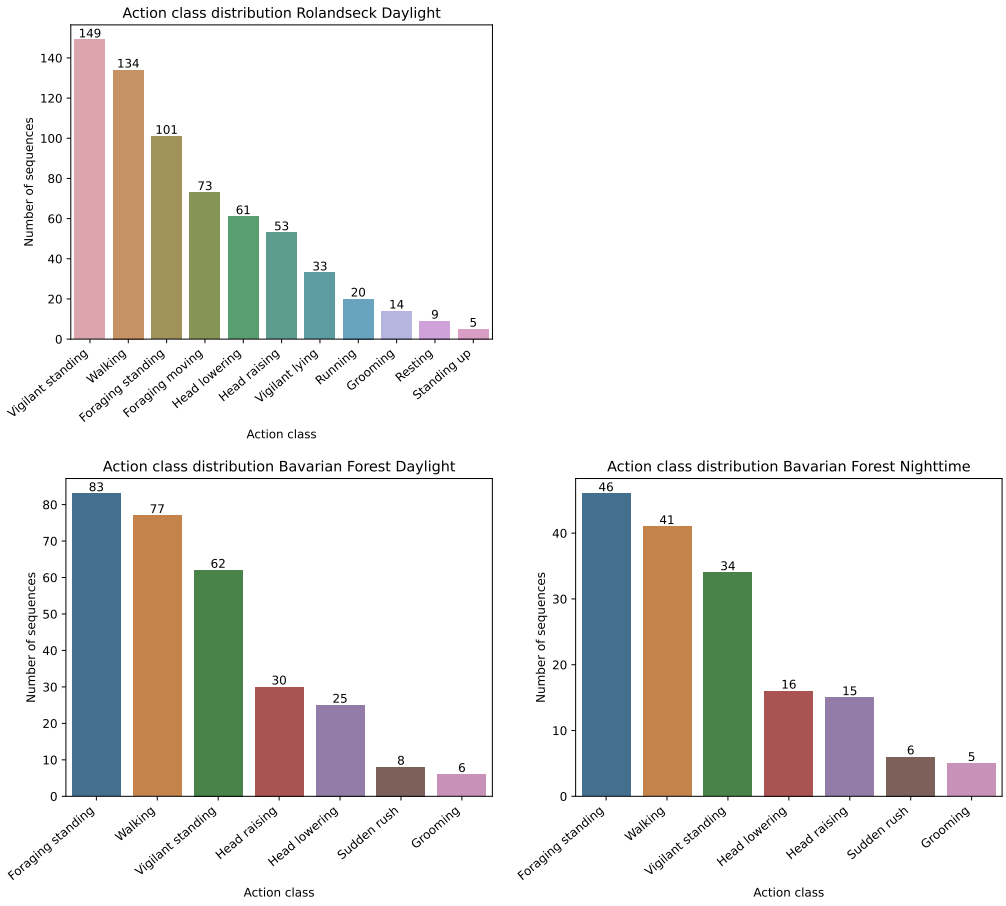

Figure 2. Action class distribution of the three datasets.

3.2. Action Detection

In this section, we present our novel action detection system. We show the workflow of our approach in Figure 3. First, our instance segmentation system SWIFT [21] detects and tracks all animals in the video data. This represents the detection part of the action detection. Based on the bounding boxes and the track IDs of the detected animals, we cut them out from the video. These cutouts are always square so that they fit the following action recognition and are not compressed or stretched. After that, we use the instance masks found by SWIFT to cut out the exact contour of the animal. This is especially important if several animals are close to each other and thus several individuals appear within the cutout. But also in general it helps to cut out the animal, because this creates an

independence from the background of the animal and the system can generalize better to new unseen scenes. Then, the cutouts are resized to the same size as input to our novel action recognition network MAROON. The action recognition predicts the action class resulting in the final action detection result: the action class, instance mask, bounding box, (animal) class, score and track ID.

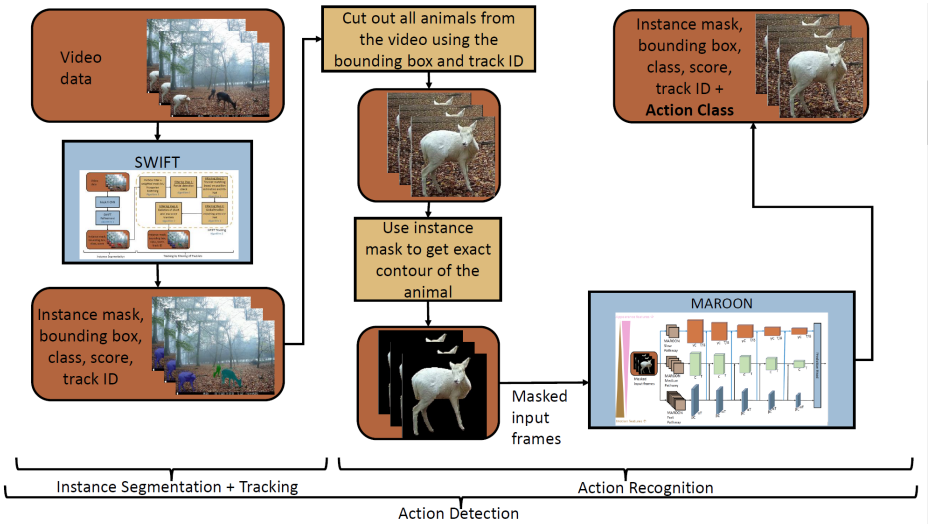

Figure 3. Our action detection system, a combination of SWIFT and MAROON. Brown boxes represent data, blue boxes describe neural networks and algorithms and yellow boxes represent algorithmic instructions.

3.2.1. Action Recognition with MAROON

Here, we explain in detail the architecture and idea of our action recognition network MAROON—mask-guided action recognition. In Figure 4, we show the architecture of MAROON. We highlight in blue the pathways and lateral connections that we newly introduce in comparison to the base model of SlowFast [22]. The two most important innovations in our network are (1) the use of masked input frames and (2) the triple-stream approach. As described previously, the input frames of the action recognition network are first cut out from the overall video frame using the bounding boxes detected by SWIFT and then masked by the instance mask of the animal. This allows MAROON to focus entirely on the actor during the feature extraction and not learn unnecessary background information.

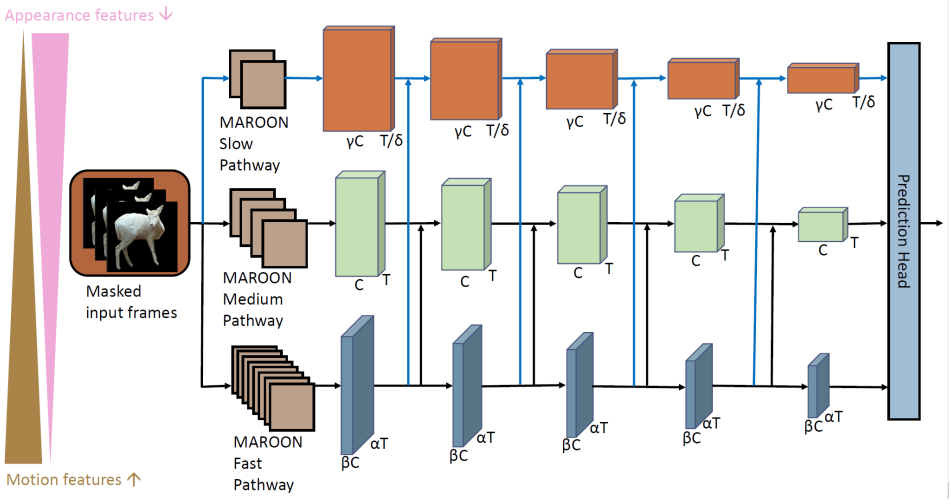

Figure 4. Our action recognition network MAROON uses three streams and masked input frames. The blue arrows represent the new third path and the new lateral connections in the MAROON architecture compared to SlowFast.

Our network architecture builds on the idea of SlowFast [22]. SlowFast introduces a two-stream network, where two pathways exist, a slow pathway and a fast pathway. The task of the fast pathway is to extract motion features (for example, the type of movement and speed of the animal), and the slow pathway should concentrate on the appearance features (for example, the color and the pose of the animal). To achieve this, they modify the number of frames T that are fed into each of the two streams and the number of feature channels C that the convolutional layers have in the respective streams. For this, the parameters α and β are introduced. For MAROON, we extend this idea from a two-stream approach to a triple-stream approach to extract motion and appearance features with different granularities. In Figure 4, we show the architecture of MAROON. Accordingly, we name the three pathways the MAROON slow pathway, MAROON medium pathway, and MAROON fast pathway. We newly introduce the parameters γ and δ. With α and δ ($\alpha, \delta > 1$), we steer the amount of frames for each of the three pathways. The MAROON fast pathway receives the most densely sampled αT frames. The MAROON medium pathway receives T frames, and the MAROON slow pathway receives only T/δ frames. A typical distribution of frames, for example, if you consider a 16-frame input sequence, would be 16 frames for the fast pathway, 4 frames for the medium pathway, and 2 frames for the slow pathway. The second aspect in which the three pathways differ is the channel capacity C, which is controlled by the two parameters β and γ. A higher amount of channels allows for the extraction of more detailed features. This is the case when the appearance of the animal is in focus. The medium pathway has C channels. The parameters β and γ ($\beta < 1, \gamma > 1$) steer the amount of channels for the other two pathways. With βC channels, the fast pathway has less channels than the medium pathway to concentrate on the motion feature extraction. The slow pathway has with γC channels the highest number of channels of all three pathways. To summarize, the higher the number of frames fed into the pathway, the lower the channel capacity and vice versa.

In addition to the two-stream architecture, the connection of both streams by lateral connections is one of the crucial innovations of SlowFast [22]. Thus, the features extracted from the fast pathway are regularly combined with the features of the other pathway. We extend this idea to our third pathway by combining the features of the fast pathway with the features of our slow pathway in the same way. The features from the three pathways are combined in the prediction head that generates the final action class prediction.

4. Results

In this section, we show our evaluation results of MAROON and compare them with the state-of-the-art approaches SlowFast [22] and MViT [23]. Moreover, we perform ablation studies to prove our architecture's functionality and the choice of our parameters.

4.1. Training and Testing Details

We choose SlowFast as a comparison model because it is a very successful CNN approach and in particular because MAROON is inspired by SlowFast and extends upon its ideas. MViT is a very prominent transformer model that performs well in action recognition on person data. We choose the parameters for these models following the values proposed by the respective papers and through various model tests.

We base the maximum length of the input sequences on the action classes, which describe temporally short actions such as head raising or a sudden rush. For the Rolandseck Daylight dataset, we choose an input length of 16 frames. Since this dataset has a high FPS of 30, fast actions are also well covered. For the Bavarian Forest Daylight and Bavarian Forest Nighttime datasets, we have to limit ourselves to eight frames.

As shown in Section 3, the action class distributions in our datasets are long-tailed distributions. This means that the underrepresented classes have to be specifically considered to achieve good predictions due to the low number of samples during training. To achieve better action recognition results on underrepresented classes, we perform oversampling during the training process. In each epoch, we randomly sample from the smaller classes, so

that each class represents at least 75% of the number of observations of the largest class. We determine this parameter through experiments. We restrict the resampling to 75% because our smallest classes only contain 5 samples and the largest class more than 100 samples. Even with data augmentation strategies, these samples are presented to the network very frequently. With this restriction, we try to prevent overfitting on the individual samples.

During training, we use random horizontal flips as data augmentation. Moreover, we perform temporal and spatial jittering. In general, the action sequences are longer than the exact input length for the networks, in our case, 16 or 8 frames. For temporal jittering, this desired sequence of fixed length is randomly sampled from the whole action sequence. With temporal jittering, the model receives a different part of the entire action sequence each epoch, avoiding overfitting. As described before, we cut out each animal by their bounding box. For spatial jittering, the box is cut out a little bit larger than the desired input size for the network. Then, the correct size is cut out randomly from this larger patch. With this technique, the animal is not always at the center position and therefore there is more variety in the training data. For comparable testing, the input must always be the same. Accordingly, spatial jittering is not performed. However, in order to cover several points in time of an action sequence, the sequence is divided into equal sections as test sequences. This is called ensemble view testing. We use 10-view testing.

For evaluating the action recognition quality of networks, we use the common metric of top-1 and top-5 accuracy. In general, top-k accuracy means that there is a correct prediction for a sample if the k best predictions of the network include the correct action class of the sample. There is no common metric for the task of action detection. However, the mean average precision (mAP) is sometimes used (for example, in [52,53]). One problem with this metric is that it only takes into account the bounding box accuracy of the detector (and not instance masks, for example) and therefore creates a high dependency of the action class prediction on the detection. By evaluating the detector and the action recognition approach separately, both parts are evaluated more fairly, and at the same time it is easier to decide whether a part of the overall system should be replaced.

We train all our models for 200 epochs. To ensure a fair comparison among the models, we do not use pretrained weights for initialization. For MAROON and also for SlowFast, we use the ResNet-50 as the backbone. For the Rolandseck Daylight dataset, the input sequence length is 16 frames. We set the parameters for MAROON for all datasets as $\alpha = 4, \beta = 1/8, \gamma = 4$ and $\delta = 8$. We determine these parameters through extensive experiments. The parameters α and δ lead to the input sizes of 16 frames for the fast pathway, 4 frames for the medium pathway, and 2 frames for the slow pathway for the Rolandseck Daylight dataset and 8 frames for the fast pathway, 2 frames for the medium pathway, and 1 frame for the slow pathway for the Bavarian Forest datasets.

4.2. Evaluation Results

We evaluate our models on all three presented datasets. We divide our datasets into train and test sets, so that about 20% of each class is included in the test set. Due to the small amount of data, we do not create a validation set. However, to show that our model generally performs better than the comparison models, we perform a stratified 5-fold cross-validation on all datasets. Stratified means that 20% of the data per class are selected as test data in each fold. In this way, each test set represents the action class distribution of the whole dataset. In Table 2, we show our evaluation results for the different models and datasets.

MAROON outperforms the other models for all three datasets for the cross-validation. On average, all models perform best on the Rolandseck Daylight dataset and worst on the Bavarian Forest Nighttime dataset. This can be explained by the fact that the Rolandseck Daylight dataset contains the most sequences and can therefore create the most diversity of situations in training, whereas the Bavarian Forest Nighttime dataset contains the fewest sequences. In addition, the nighttime dataset is more difficult than daytime datasets due to

the lack of color information as infrared flashes are used for recording. The accuracies of the two comparison models are approximately the same.

Table 2. Evaluation results of our model MAROON compared for the different Rolandseck Daylight, Bavarian Forest Daylight and Bavarian Forest Nighttime datasets with 5-fold cross-validation. The top-1 and top-5 accuracies are depicted. The best value for each dataset is marked in bold.

Dataset	Model	Top-1	Top-5
Rolandseck Daylight	MAROON	**69.16**	**96.31**
	SlowFast	42.05	89.66
	MViT	43.13	85.18
Bavarian Forest Daylight	MAROON	**46.39**	**97.24**
	SlowFast	35.40	95.88
	MViT	35.05	94.17
Bavarian Forest Nighttime	MAROON	**43.05**	**96.33**
	SlowFast	35.49	95.68
	MViT	31.91	93.26

In Figure 5, we show the top-1 accuracies for all action classes of the 5-fold cross-validation. In general, MAROON outperforms the other two models for all classes in all three datasets. There are only a few exceptions in the Bavarian Forest Daylight and Bavarian Forest Nighttime datasets for classes with few samples. MAROON achieves similar top-1 accuracies for all action classes, which is especially visible in the Rolandseck Daylight dataset. This ensures that the model does not focus on action classes with many samples but also considers smaller action classes.

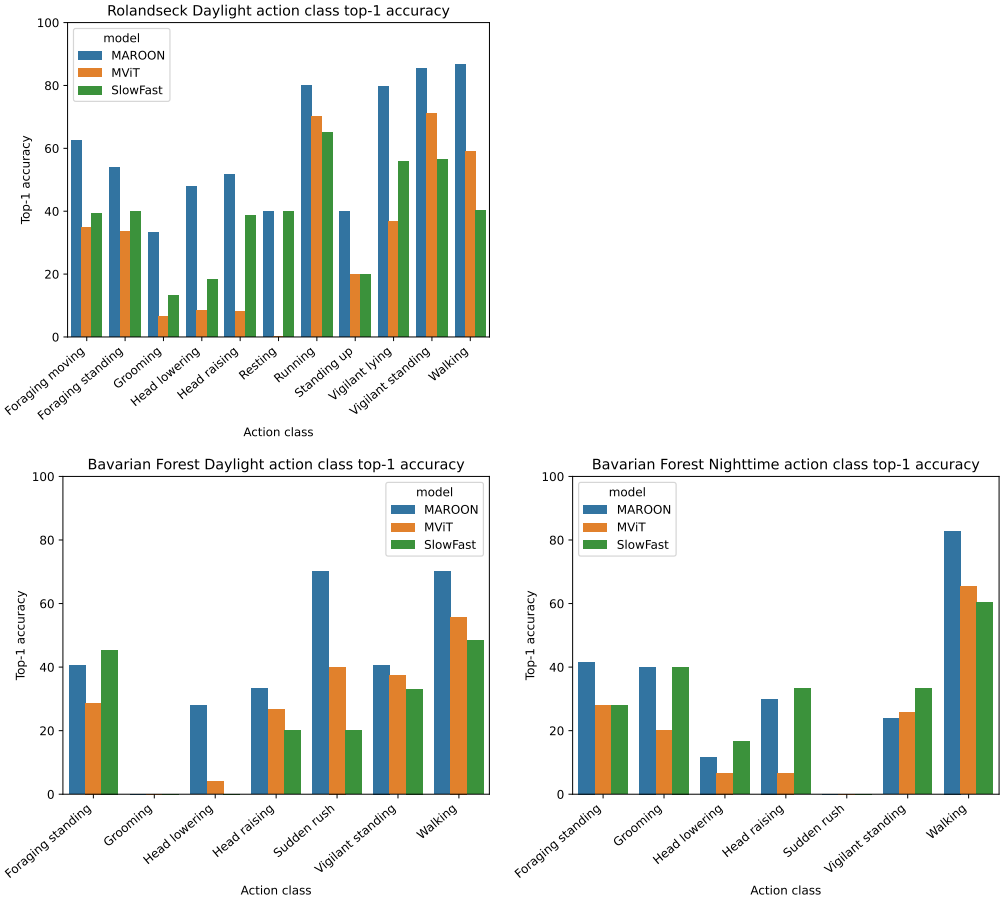

Figure 5. Action class top-1 accuracy of the cross-validation of the three datasets with the models MAROON, MViT and SlowFast.

4.3. Ablation Experiments

In this section, we evaluate the functionality of the different parts of our action recognition network. We use the same 5-fold cross-validation as in the section before.

In Table 3, we show the impact of the masked input. We use the masked input frames as input for SlowFast and MViT and also use regular (non-masked) input frames for MAROON. The mask information improves the results for all models on all datasets by 10 percentage points on average. However, our action recognition network MAROON achieves the best (or at least an equally good) result compared to the other approaches, both with and without masks. Particularly, for the Rolandseck Daylight dataset, where the mask information has the greatest influence, the result of MAROON is more than 11 percentage points better than those of SlowFast and MViT. On all datasets, the MViT transformer model benefits the least relatively from the addition of the mask information. This can be explained by the way transformers work. Transformers use an attention mechanism and therefore look at small image sections. The masking results in larger, uniformly black-colored areas that look the same or contain only little image information of the animal when entered into the transformer.

Table 3. Analysis of the impact of masked input frames. The best value for each dataset is marked in bold.

Dataset	Model	Top-1 (without Mask)	Top-5 (without Mask)	Top-1 (with Mask)	Top-5 (with Mask)
Rolandseck Daylight	MAROON	54.65	90.96	**69.19**	**96.31**
	SlowFast	42.05	89.66	57.67	92.48
	MViT	43.13	85.18	49.41	89.57
Bavarian Forest Daylight	MAROON	36.44	94.85	**46.39**	**97.24**
	SlowFast	35.40	95.88	45.35	96.57
	MViT	35.05	94.17	42.63	95.54
Bavarian Forest Nighttime	MAROON	35.04	91.42	**43.05**	**96.33**
	SlowFast	35.49	95.68	42.93	93.86
	MViT	31.91	93.26	40.62	92.03

Furthermore, we analyze the importance of the lateral connection between the MAROON slow pathway and the MAROON fast pathway. In Table 4, we show the results for different possible connections. The authors of [22] also determine the importance of the lateral connection between their SlowFast slow pathway and SlowFast fast pathway. In our evalution, we consider four different possibilites: the connection between the fast and slow pathways (that is the final choice for MAROON), the connection between the medium and slow pathways (before and after merging with the fast pathway), and no connection at all. For the Rolandseck Daylight and Bavarian Forest Nighttime datasets, the fast to slow connection performs best. For the Bavarian Forest Daylight dataset, the fast to slow connection is the second-best choice after the medium (before merging) to slow connection. Overall, the introduction of the third pathway in particular seems to be helpful, and the type of connection is not so decisive. However, since the no connection option is never the best choice, combining the features from the other streams with the new MAROON slow pathway seems to be a promising option. This also corresponds to our theoretical considerations in Section 3.2.1.

Table 4. Analysis of the lateral connections to the new MAROON slow pathway. The best value for each dataset is marked in bold.

Dataset	Type of Lateral Connection	Top-1	Top-5
Rolandseck Daylight	fast to slow	**69.16**	**96.31**
	no connection	65.50	95.70
	medium (before merging) to slow	66.74	94.32
	medium (after merging) to slow	68.26	96.00
Bavarian Forest Daylight	fast to slow	46.39	97.24
	no connection	45.71	**97.94**
	medium (before merging) to slow	**51.89**	96.21
	medium (after merging) to slow	46.38	96.92
Bavarian Forest Nighttime	fast to slow	**43.05**	96.33
	no connection	42.35	**96.93**
	medium (before merging) to slow	39.18	95.10
	medium (after merging) to slow	41.70	95.09

5. Discussion

Our evaluation results show that MAROON outperforms other state-of-the-art approaches for the different datasets. Our three datasets cover various situations and scenes. In the Rolandseck Daylight dataset, groups of animals are often present, where a distinction of individual animals through instance masks is very important to obtain reliable action predictions. The Bavarian Forest Daylight dataset includes videos with only a few animals and additionally has a lower resolution and FPS rate. The Bavarian Forest Nighttime dataset shows that MAROON is also able to predict actions on videos without color information and with a very low FPS rate. In general, an easy solution to improve the accuracy of all models for the different datasets would be to increase the number of videos and therefore increase the data material for training and testing. However, the annotation process of the videos is very time-consuming and rare action classes are often underrepresented in the data. It is a common phenomenon in wildlife ecology [18] and in computer vision as a whole [57] that action classes occur very unevenly and are represented by long-tailed distributions. If the (very) rare action classes could be supplemented by further examples, this would improve the accuracy of the models. However, this is limited not only by the annotation but also by the rareness of the camera trap recordings of these actions. Often, these actions are of special interest to researchers. Therefore, we apply oversampling, one of the most widely used techniques to train long-tailed distributions more efficiently. Due to the very small number of some actions, undersampling is not useful in order not to restrict the data diversity for the more frequent classes too much. In order to generate a larger variety of data during training, we use the data augmentation techniques already explained in Section 4.1, such as horizontal flips, temporal jittering and spatial jittering.

If sufficient time and resources are available for a study, it is also possible to make additional video recordings in zoos or wild animal enclosures in order to generate data material for rare behavior. Another way to supplement data from rare behaviors is to generate artificial data. Even if the results of today's artificially generated images are already very high, it is still problematic to transfer a network trained on artificially generated images to real images when considering such complex scenes as wildlife videos with changing weather conditions, different exposures and day and night recordings. However, with further optimization of the generation of artificially generated data material, this is a promising possibility to supplement the data material in the future.

Our action recognition network is limited to the detection of actions of single animals, i.e., no interactions between animals can be detected. In our recorded data, this case was extremely rare and therefore neglected. Furthermore, most of the action recognition approaches concentrate on the identification of actions of one actor. But there are certainly other species and datasets where actions between different animal individuals are also of interest. The action detection approaches as shown in [51,52] determine actions between

humans. Our approach would need to be adapted so that multiple animals are masked as input and entered into the action recognition network to also describe interactions between animals.

The application of our action detection system to other animal species, especially to other similarly sized animals, is easy to implement in principle. Data material for the desired animal species and the corresponding actions must first be available and annotated in the same way. SWIFT and MAROON must then be trained for the new animal species and action classes. As the detection by the Mask R-CNN in SWIFT is very reliable and can be used for many objects, a transfer is easily possible in most cases. If small animals such as birds are to be examined, it must be taken into account that the animals may only be represented by a very small region within the video (e.g., a size of less than 50×50 pixels). This can be problematic for the detection and even more problematic for action recognition due to the missing visual details. Care should therefore be taken to ensure that a correct camera setting is used when recording data.

In general, the resolution of the video recordings is important for recognizing the animals. It is easier for instance segmentation with SWIFT to segment more object details if the animal is closer to the camera and therefore has more pixels. As a result, a better image resolution also means easier recognition of the action classes in the action recognition part. However, the action recognition ability also depends on the action classes under consideration, i.e., how many details of the animal are necessary to recognize the action and whether there are similar actions with which it could be confused. If our approach is to be transferred to other animal species, the video resolution must also be taken into account. If our system is to be used for smaller animals or for birds, for example, they would probably be too small in our camera setting to distinguish between several actions. The recordings from the Rolandseck dataset already have a very high resolution. It is difficult to increase the video quality further because of storage restrictions on the camera trap devices. Therefore, the recording settings should be adapted to the desired animal classes and action classes. The recording setting (including the proximity of the camera to the animals or the location of the recording) should be adjusted accordingly.

Extending the idea of SlowFast [22] and using three pathways for MAROON improves the action recognition accuracy. A drawback from the expansion of the model is that it now has more parameters than SlowFast and therefore takes more time to train. Thus, if very large datasets are used, either longer training times must be scheduled or more hardware must be used for the training process.

The introduction of masked input sequences has improved our action prediction accuracy by an average of 10 percentage points. However, this also means that an instance mask must be available for the different animals. Action recognition datasets usually contain only video sequences focused on the actor. Action detection datasets usually use only bounding boxes and no instance masks to localize the different actors, for example, the person action detection dataset AVA [58]. Since we use SWIFT as a reliable instance segmentation and tracking tool in our action detection system, an application to fallow deer, red deer and roe deer is not problematic. However, if other animal species or humans are to be examined, SWIFT must first be trained with appropriate instance masks. If no dataset with instance masks is available, this means that annotations with instance masks must be created for the desired new animal species. In principle, this is more time-consuming than an annotation with bounding boxes. However, as annotation tools with AI support are constantly being developed further [56,59] and foundation models such as Segment Anything [60] are emerging at the same time, this may become an easier task in the future than it is today.

In Section 4.2, we present the class-specific action recognition accuracies for our models and datasets. There are two reasons why both our MAROON approach and the other two action recognition networks achieve lower accuracies on some classes than on others. One simple reason is that some classes (like resting, standing up or a sudden rush) occur less frequently than others, as already discussed. Even if this problem is partially compensated

for by the introduced oversampling, it may still happen that the training data are not diverse enough to generalize successfully. A targeted enlargement of the dataset with regard to these classes (given that there are sufficient recordings that show this behavior) is therefore a possibility to improve the prediction accuracy. The second reason why some action classes are difficult to predict is that they can be performed differently and also may be similar to other classes. Examples of this are the classes foraging moving and walking. Foraging moving differs from the action class walking only in the aspect that the animal searches for food. For example, it is characteristic that the animal's head is lowered in comparison to walking. The action classes foraging standing and vigilant standing can also be very similar, as the animal's head is not lowered, particularly when it is searching for food, for example, at a bush. Here, too, increasing the amount of samples can make it easier to distinguish between the classes. But, in general, it can be seen that the action recognition accuracies for these classes are higher than for the classes belonging to the first reason mentioned.

When an action detection system is applied to a video where animals change their behavior (which is a normal case), there are transition areas between two behaviors. In these, the prediction changes from one behavior to the next. The findings of the analysis of behavioral sequences from ethology [61–63] could be helpful in the future to support the action prediction process. In addition, these studies can also be used for persons [64]. For example, there are behaviors that are more likely to follow other behaviors. This initially has no effect on the evaluation of the action recognition approaches, as only one sequence in which only one behavior occurs is considered here. However, this could be helpful when applying the system to unseen recordings. This additional information could make the prediction by action recognition even more accurate or verify its results.

6. Conclusions

In this work, we present our novel action detection system that can be applied within wildlife ecology, especially the new action recognition network MAROON. In detail, we combine our already successful instance segmentation and tracking system SWIFT with MAROON to perform action detection. We evaluate the functionality of MAROON on three different wildlife datasets containing fallow deer, roe deer and red deer. MAROON is the first action recognition approach that combines instance mask information for the input sequences with a feature extraction in a triple-stream approach for action recognition. The automated action detection of animals forms an important basis for behavioral studies.

In our experiments, we have shown that MAROON improves the action recognition accuracy on all three datasets compared to other state-of-the-art approaches. For the Rolandseck Daylight dataset, MAROON achieves a top-1 accuracy of 69.19% in comparison to 43.13% from MViT and 42.05% from SlowFast. Also, on the Bavarian Forest datasets our action recognition network achieves a 10 percent point higher top-1 accuracy compared to the other approaches. We perform a 5-fold cross-validation to show that our approach generally works better than the other approaches.

We believe that the use of our action detection approach in wildlife ecology will be beneficial for ecologists by eliminating the need to analyze all video data material from camera traps visually while enabling new insights through application to even larger datasets. In the future, we plan to evaluate our action detection system on more datasets showing different action classes. Moreover, we plan to expand our action recognition network so that it can also describe interactions between animals. However, this will first require appropriate data material.

Author Contributions: Conceptualization, F.S., S.v.B.C., M.H. and V.S.; methodology, F.S., S.v.B.C., M.H. and V.S.; software, F.S.; validation, F.S. and V.S.; formal analysis, F.S.; investigation, F.S., S.v.B.C., M.H. and V.S.; resources, F.S., V.S., S.v.B.C. and M.H.; data curation, F.S. and S.v.B.C.; writing—original draft preparation, F.S.; writing—review and editing, F.S., S.v.B.C., M.H. and V.S.; visualization, F.S.; supervision, V.S. and M.H.; project administration, V.S.; funding acquisition, V.S. and M.H. All authors have read and agreed to the published version of this manuscript.

Funding: This work was partially conducted within the project "Automated Multisensor station for Monitoring Of species Diversity" (AMMOD), which is funded by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung (BMBF), Bonn, Germany (FKZ 01LC1903B).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: We would like to thank the other member of our working group, Florian Huber, for fruitful discussions on aspects of this study. We especially want to thank the student Hannah Hasert for annotating a main part of the Bavarian Forest Daylight and Bavarian Forest Nighttime datasets. For helping to annotate the Rolandseck dataset we also want to thank the students Iva Ewert, Alvin Inderka, Bertan Karacora, Stefan Messerschmidt, Jennifer Collard, Pablo Rauh Corro and Simon Mathy, who worked on student projects in our Intelligent Vison Systems (IVSs) group and thereby annotated video data of our dataset.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Description of Action Classes

We briefly define the action classes that we use in this study:

Foraging moving: The animal is moving or walking while the head is held towards the ground or towards a food source (bushes, young trees) with eyes opened. The mouth is near (and eventually touching) the ground or the food source, and the jaws are eventually moving.

Foraging standing: The animal is standing while the head is held towards the ground or towards a food source (bushes, young trees) with eyes opened. The mouth is near (and eventually touching) the ground or the food source, and the jaws are eventually moving.

Grooming: The animal is standing while scratching or licking itself. The animal's mouth is touching a random body part while the head is moving slightly.

Head lowering: The animal's head is moved from the raised (parallel line with the body or higher) position to the lowered (head nearer towards the ground up to nearly parallel line with the body) position within a short time period.

Head raising: The animal's head is moved from the lowered (head nearer towards the ground up to nearly parallel line with the body) position to the raised (parallel line with the body or higher) position within a short time period.

Resting: The animal's body is on the ground. The torso is in a slightly lateral position. At the same time, the legs can be stretched off the body or bent underneath and besides the body. The head is placed down on the ground. The animal's eyes can be either closed or opened.

Running: A faster target-oriented forward movement than walking. The posture is relatively strained with the animal's head raised and eyes opened. Two or more hooves do not touch the ground. It contains everything from trot to full speed.

Sudden rush: The animal goes from standing to running without walking in between and within 1 s.

Standing up: The animal changes its position from lying to standing.

Vigilant lying: Lying with the head held high and occasional turning of the head and with ear twitching.

Vigilant standing: Standing with a strained posture and with the head held parallel to body or higher. The animal is looking around and/or twitching the ears occasionally.

Walking: A relatively slow, target-oriented forward movement while not feeding or chewing. The posture is relaxed with the animal's head parallel to the body or higher and eyes opened. One hoof does not touch the ground.

References

1. Berger-Tal, O.; Polak, T.; Oron, A.; Lubin, Y.; Kotler, B.P.; Saltz, D. Integrating animal behavior and conservation biology: A conceptual framework. *Behav. Ecol.* **2011**, *22*, 236–239. [CrossRef]
2. Caravaggi, A.; Banks, P.B.; Burton, A.C.; Finlay, C.M.; Haswell, P.M.; Hayward, M.W.; Rowcliffe, M.J.; Wood, M.D. A review of camera trapping for conservation behaviour research. *Remote Sens. Ecol. Conserv.* **2017**, *3*, 109–122. [CrossRef]
3. McCallum, J. Changing use of camera traps in mammalian field research: Habitats, taxa and study types. *Mammal Rev.* **2013**, *43*, 196–206. [CrossRef]
4. Wearn, O.R.; Glover-Kapfer, P. Snap happy: Camera traps are an effective sampling tool when compared with alternative methods. *R. Soc. Open Sci.* **2019**, *6*, 181748. [CrossRef] [PubMed]
5. Hongo, S.; Nakashima, Y.; Yajima, G.; Hongo, S. A practical guide for estimating animal density using camera traps: Focus on the REST model. *bioRxiv* **2021**. . [CrossRef]
6. Villette, P.; Krebs, C.J.; Jung, T.S. Evaluating camera traps as an alternative to live trapping for estimating the density of snowshoe hares (*Lepus americanus*) and red squirrels (*Tamiasciurus hudsonicus*). *Eur. J. Wildl. Res.* **2017**, *63*, 1–9. [CrossRef]
7. Henrich, M.; Burgueño, M.; Hoyer, J.; Haucke, T.; Steinhage, V.; Kühl, H.S.; Heurich, M. A semi-automated camera trap distance sampling approach for population density estimation. *Remote Sens. Ecol. Conserv.* **2023**. . [CrossRef]
8. Tobler, M.; Carrillo-Percastegui, S.; Leite Pitman, R.; Mares, R.; Powell, G. Further notes on the analysis of mammal inventory data collected with camera traps. *Anim. Conserv.* **2008**, *11*, 187–189. [CrossRef]
9. Linkie, M.; Dinata, Y.; Nugroho, A.; Haidir, I.A. Estimating occupancy of a data deficient mammalian species living in tropical rainforests: Sun bears in the Kerinci Seblat region, Sumatra. *Biol. Conserv.* **2007**, *137*, 20–27. [CrossRef]
10. Frey, S.; Fisher, J.T.; Burton, A.C.; Volpe, J.P. Investigating animal activity patterns and temporal niche partitioning using camera-trap data: Challenges and opportunities. *Remote Sens. Ecol. Conserv.* **2017**, *3*, 123–132. [CrossRef]
11. Caravaggi, A.; Zaccaroni, M.; Riga, F.; Schai-Braun, S.C.; Dick, J.T.; Montgomery, W.I.; Reid, N. An invasive-native mammalian species replacement process captured by camera trap survey random encounter models. *Remote Sens. Ecol. Conserv.* **2016**, *2*, 45–58. [CrossRef]
12. Green, S.E.; Rees, J.P.; Stephens, P.A.; Hill, R.A.; Giordano, A.J. Innovations in camera trapping technology and approaches: The integration of citizen science and artificial intelligence. *Animals* **2020**, *10*, 132. [CrossRef] [PubMed]
13. Mitterwallner, V.; Peters, A.; Edelhoff, H.; Mathes, G.; Nguyen, H.; Peters, W.; Heurich, M.; Steinbauer, M.J. Automated visitor and wildlife monitoring with camera traps and machine learning. *Remote Sens. Ecol. Conserv.* **2023**. . [CrossRef]
14. Vélez, J.; McShea, W.; Shamon, H.; Castiblanco-Camacho, P.J.; Tabak, M.A.; Chalmers, C.; Fergus, P.; Fieberg, J. An evaluation of platforms for processing camera-trap data using artificial intelligence. *Methods Ecol. Evol.* **2023**, *14*, 459–477. [CrossRef]
15. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *Adv. Neural Inf. Process. Syst.* **2014**, *27* , 568—576.
16. Kong, Y.; Fu, Y. Human action recognition and prediction: A survey. *Int. J. Comput. Vis.* **2022**, *130*, 1366–1401. [CrossRef]
17. Schindler, F.; Steinhage, V. Identification of animals and recognition of their actions in wildlife videos using deep learning techniques. *Ecol. Inform.* **2021**, *61*, 101215. [CrossRef]
18. Sakib, F.; Burghardt, T. Visual recognition of great ape behaviours in the wild. *arXiv* **2020**, arXiv:2011.10759.
19. Ng, X.L.; Ong, K.E.; Zheng, Q.; Ni, Y.; Yeo, S.Y.; Liu, J. Animal kingdom: A large and diverse dataset for animal behavior understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 19023–19034.
20. Brookes, O.; Mirmehdi, M.; Kühl, H.; Burghardt, T. Triple-stream Deep Metric Learning of Great Ape Behavioural Actions. *arXiv* **2023**, arXiv:2301.02642.
21. Schindler, F.; Steinhage, V. Instance segmentation and tracking of animals in wildlife videos: SWIFT-segmentation with filtering of tracklets. *Ecol. Inform.* **2022**, *71*, 101794. [CrossRef]
22. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6202–6211.
23. Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; Feichtenhofer, C. Multiscale vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6824–6835.
24. Zhang, H.B.; Zhang, Y.X.; Zhong, B.; Lei, Q.; Yang, L.; Du, J.X.; Chen, D.S. A comprehensive survey of vision-based human action recognition methods. *Sensors* **2019**, *19*, 1005. [CrossRef] [PubMed]
25. Bhoi, A. Spatio-temporal action recognition: A survey. *arXiv* **2019**, arXiv:1901.09403.
26. Liu, X.; Bai, S.; Bai, X. An empirical study of end-to-end temporal action detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 20010–20019.
27. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
28. Carreira, J.; Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6299–6308.
29. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6450–6459.

30. Hara, K.; Kataoka, H.; Satoh, Y. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6546–6555.
31. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
32. Zhou, B.; Andonian, A.; Oliva, A.; Torralba, A. Temporal relational reasoning in videos. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 803–818.
33. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal segment networks for action recognition in videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2740–2755. [CrossRef] [PubMed]
34. Feichtenhofer, C. X3d: Expanding architectures for efficient video recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 203–213.
35. Christoph, R.; Pinz, F.A. Spatiotemporal residual networks for video action recognition. *Adv. Neural Inf. Process. Syst.* **2016**, *2*, 3468–3476.
36. Sheth, I. Three-stream network for enriched Action Recognition. *arXiv* **2021**, arXiv:2104.13051.
37. Liang, D.; Fan, G.; Lin, G.; Chen, W.; Pan, X.; Zhu, H. Three-stream convolutional neural network with multi-task and ensemble learning for 3d action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
38. Jiang, B.; Wang, M.; Gan, W.; Wu, W.; Yan, J. Stm: Spatiotemporal and motion encoding for action recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2000–2009.
39. Lin, J.; Gan, C.; Han, S. Tsm: Temporal shift module for efficient video understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7083–7093.
40. Li, Y.; Ji, B.; Shi, X.; Zhang, J.; Kang, B.; Wang, L. Tea: Temporal excitation and aggregation for action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 909–918.
41. Kwon, H.; Kim, M.; Kwak, S.; Cho, M. Motionsqueeze: Neural motion feature learning for video understanding. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XVI 16; Springer: Cham, Switzerland, 2020; pp. 345–362.
42. Ryoo, M.S.; Piergiovanni, A.; Arnab, A.; Dehghani, M.; Angelova, A. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv* **2021**, arXiv:2106.11297.
43. Chen, M.; Wei, F.; Li, C.; Cai, D. Frame-wise action representations for long videos via sequence contrastive learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13801–13810.
44. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998—6008.
45. Yan, S.; Xiong, X.; Arnab, A.; Lu, Z.; Zhang, M.; Sun, C.; Schmid, C. Multiview transformers for video recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3333–3343.
46. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; Schmid, C. Vivit: A video vision transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada., 11–17 October 2021; pp. 6836–6846.
47. Sushmit, A.S.; Ghosh, P.; Istiak, M.A.; Rashid, N.; Akash, A.H.; Hasan, T. SegCodeNet: Color-Coded Segmentation Masks for Activity Detection from Wearable Cameras. *arXiv* **2020** arXiv:2008.08452.
48. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
49. Zaghbani, S.; Bouhlel, M.S. Mask rcnn for human motion and actions recognition. In *Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020)*; Springer: Cham, Switzerland, 2021; pp. 1–9.
50. Hacker, L.; Bartels, F.; Martin, P.E. Fine-Grained Action Detection with RGB and Pose Information using Two Stream Convolutional Networks. *arXiv* **2023**, arXiv:2302.02755.
51. Tang, J.; Xia, J.; Mu, X.; Pang, B.; Lu, C. Asynchronous interaction aggregation for action detection. In *Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020*; Proceedings, Part XV 16; Springer: Cham, Switzerland, 2020; pp. 71–87.
52. Biswas, S.; Gall, J. Discovering Multi-Label Actor-Action Association in a Weakly Supervised Setting. In Proceedings of the Asian Conference on Computer Vision, Online, 30 November–4 December 2020.
53. Chen, L.; Tong, Z.; Song, Y.; Wu, G.; Wang, L. Efficient Video Action Detection with Token Dropout and Context Refinement. *arXiv* **2023**, arXiv:2304.08451.
54. Yuan, L.; Zhou, Y.; Chang, S.; Huang, Z.; Chen, Y.; Nie, X.; Wang, T.; Feng, J.; Yan, S. Toward accurate person-level action recognition in videos of crowed scenes. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 4694–4698.
55. Dutta, A.; Zisserman, A. The VIA Annotation Software for Images, Audio and Video. In Proceedings of the 27th ACM International Conference on Multimedia, MM '19, Nice, France, 21–25 October 2019. [CrossRef]

56. Sofiiuk, K.; Petrov, I.A.; Konushin, A. Reviving iterative training with mask guidance for interactive segmentation. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 3141–3145.
57. Zhang, Y.; Kang, B.; Hooi, B.; Yan, S.; Feng, J. Deep long-tailed learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**. . [CrossRef] [PubMed]
58. Gu, C.; Sun, C.; Ross, D.A.; Vondrick, C.; Pantofaru, C.; Li, Y.; Vijayanarasimhan, S.; Toderici, G.; Ricco, S.; Sukthankar, R.; et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6047–6056.
59. Liu, Q.; Xu, Z.; Bertasius, G.; Niethammer, M. Simpleclick: Interactive image segmentation with simple vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 22290–22300.
60. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment anything. *arXiv* **2023**, arXiv:2304.02643.
61. Chatfield, C.; Lemon, R.E. Analysing sequences of behavioural events. *J. Theor. Biol.* **1970**, *29*, 427–445. [CrossRef] [PubMed]
62. Bels, V.L.; Pallandre, J.P.; Pelle, E.; Kirchhoff, F. Studies of the Behavioral Sequences: The Neuroethological Morphology Concept Crossing Ethology and Functional Morphology. *Animals* **2022**, *12*, 1336. [CrossRef]
63. Gygax, L.; Zeeland, Y.R.; Rufener, C. Fully flexible analysis of behavioural sequences based on parametric survival models with frailties—A tutorial. *Ethology* **2022**, *128*, 183–196. [CrossRef]
64. Keatley, D. *Pathways in Crime: An Introduction to Behaviour Sequence Analysis*; Springer: Berlin/Heidelberg, Germany, 2018.