

## Article

# SOCRATES: Introducing Depth in Visual Wildlife Monitoring Using Stereo Vision

Timm Haucke <sup>1,\*</sup> , Hjalmar S. Kühl <sup>2,3,4</sup> and Volker Steinhage <sup>1</sup> <sup>1</sup> Institute of Computer Science IV, University of Bonn, Friedrich-Hirzebruch-Allee 8, 53115 Bonn, Germany<sup>2</sup> Senckenberg Museum for Natural History Görlitz, Senckenberg—Member of the Leibniz Association, Am Museum 1, 02826 Görlitz, Germany<sup>3</sup> International Institute Zittau, Technische Universität Dresden, Markt 23, 02763 Zittau, Germany<sup>4</sup> German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, Puschstrasse 4, 04103 Leipzig, Germany

\* Correspondence: haucke@cs.uni-bonn.de

**Abstract:** The development and application of modern technology are an essential basis for the efficient monitoring of species in natural habitats to assess the change of ecosystems, species communities and populations, and in order to understand important drivers of change. For estimating wildlife abundance, camera trapping in combination with three-dimensional (3D) measurements of habitats is highly valuable. Additionally, 3D information improves the accuracy of wildlife detection using camera trapping. This study presents a novel approach to 3D camera trapping featuring highly optimized hardware and software. This approach employs stereo vision to infer the 3D information of natural habitats and is designated as StereO CameRA Trap for monitoring of biodivErSity (SOCRATES). A comprehensive evaluation of SOCRATES shows not only a 3.23% improvement in animal detection (bounding box mAP<sub>75</sub>), but also its superior applicability for estimating animal abundance using camera trap distance sampling. The software and documentation of SOCRATES is openly provided.

**Keywords:** stereo vision; camera trapping; animal density; animal abundance; instance segmentation



**Citation:** Haucke, T.; Kühl, H.S.; Steinhage, V. SOCRATES: Introducing Depth in Visual Wildlife Monitoring Using Stereo Vision. *Sensors* **2022**, *22*, 9082. <https://doi.org/10.3390/s22239082>

Academic Editors: Dongyi Wang and Lirong Xiang

Received: 19 October 2022

Accepted: 18 November 2022

Published: 23 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The loss of biodiversity across a wide variety of ecosystems is accelerating. Ecologists need effective tools to monitor animal populations and address challenges such as climate and land use change, spreading of diseases, and invasive species. In recent years, camera traps have been instrumental in providing automated visual monitoring around the clock. However, all widespread camera traps are monocular, which prevents them from easily sensing their three-dimensional environment and the three-dimensional position and appearance of the observed animals.

As part of a German joint project on automated wildlife monitoring, Automated Multisensor stations for Monitoring of species Diversity (AMMOD) [1,2], we introduce SOCRATES to derive and use depth information (i.e., the distance between the camera trap and the observed scene) as a third dimension in addition to the regular two-dimensional image dimensions.

SOCRATES enables the following contributions:

- The detection and localization accuracy of animals is fostered by the additional depth information provided by SOCRATES (see Sections 3.3 and 4.2).
- Abundance estimation traditionally uses methods such as camera trap distance sampling (CTDS) with commercial camera traps, which requires laborious manual workflows. SOCRATES instead provides depth information in a fully automated way using stereo vision (see Sections 2.1 and 4).

- Reproducibility and accessibility for practitioners: The SOCRATES approach takes the practitioner's perspective and provides our raw and labeled data (see Section 3.1), code, detailed instructions, best practices, and 3D CAD models.

## 2. Related Work

Related work is reported on from three perspectives. First, the principles of stereo vision and prior work on stereo camera traps are summarized. Second, recent progress with respect to visual object detection in images and video clips is reported. Third, an overview of approaches for estimating the density and abundance of unmarked animal populations using camera traps is given.

### 2.1. Depth Using Stereo Vision

Computer stereo vision is the well-established approach to image-based depth estimation. By comparing information about an observed scene from two differing camera perspectives, depth information can be derived. Usually, both cameras in a stereo setup are displaced horizontally from one another, yielding scene observations in terms of a left image and a right image. Computer stereo vision can be seen as the technical analogue to human stereopsis, that is human perception of depth and three-dimensional structure by combining visual information from two eyes. Depth is obtained from a stereo image pair by finding the distance, or disparity, between the corresponding projections of each scene point in the left and right images. Disparity  $d$  and depth  $z$  are inversely related, i.e.,

$$z = \frac{b \cdot f}{d} \quad (1)$$

with  $b$  being the distance between both cameras (baseline) and  $f$  being the focal length of both cameras. To be able to efficiently find the projections of some scene point in both images, they first have to be rectified. Rectification results in projections of a single point to lie on a single horizontal scanline in both images and, therefore, reduces the dimensionality of the correspondence search from two to a single dimension. To obtain an accurate rectification, the intrinsic (internal camera parameters) and extrinsic (rotation and translation between the cameras) parameters have to first be obtained by a calibration procedure. For the calibration of the intrinsic parameters, a calibration object (e.g., checkerboard pattern printed on cardboard) has to be captured by the camera(s) to be able to associate 3D points in the scene with 2D points in the resulting image. To obtain the extrinsic parameters, eight or more correspondences between images of points in the projections of both cameras must be established [3]. Given a pair of rectified images, each of size  $W \times H$ , stereo matching algorithms now estimate a cost volume  $V$  of size  $W \times H \times D$ . This cost volume contains for each pixel  $(i, j)$  and for each horizontal potential disparity between 0 and the maximum expected disparity  $D - 1$  some cost measure. The cost volume might be built by using traditional techniques such as the sum-squared difference between images  $a, b$  with patch height  $c$ , i.e.,

$$V(i, j, d) = \sum_{\Delta j=-c}^c (a(i, j + \Delta j) - b(i - d, j + \Delta j))^2 \quad (2)$$

More recently,  $V$  has been estimated most successfully using learning-based approaches and convolutional neural networks [4–8]. We employed the *CREStereo* model [8] due to its demonstrated robustness in real-world conditions.

### Stereo Camera Traps

A number of works have introduced camera traps with stereo camera setups [9–11]. The hardware of [9,10] is built around an FPGA, which controls two CMOS sensors, which are mounted vertically. A pyroelectric infrared sensor is connected to a microcontroller, which is, in turn, responsible for powering on the FPGA once motion is detected. The performance of the system is evaluated with respect to absolute size estimation of artificial and

human targets. Reference [11] instead used an Intel RealSense D435 stereo camera, which computes stereo correspondence on the camera itself. However, due to the small baseline distance of roughly 5 cm, accuracy is limited at high distances. All prior works [9–11] were powered using mains electricity and were not optimized for being energy efficient and powered by a battery.

## 2.2. Instance Segmentation

State-of-the-art approaches to object detection are usually learning-based [12]. Given a 2D color or grayscale image, these methods learn to predict a set of bounding boxes, e.g., given by the 2D location of the upper-left and bottom-right corner of an axis-aligned rectangle fully enclosing the object [13]. Object detection methods might be extended to perform instance segmentation, where not only bounding boxes are predicted, but whether any pixel in the image belongs to the respective object (binary masks) [14]. Deep learning object detection and instance segmentation models usually consist of two parts. The backbone takes in the original image and produces a hierarchy of feature maps that encode higher-level information about the image. These feature maps are then used by the object detection or instance segmentation model to predict the bounding boxes and binary masks themselves [15]. A general issue of deep learning models is that their training requires immense amounts of annotated training data. Annotated data are raw data associated with corresponding labels, which can be of different modalities (e.g., object classes occurring in the image, bounding boxes around objects of interest, pixelwise masks of such objects, etc.). These labels must often be created manually and are therefore costly to obtain. This requirement of large annotated training datasets is slightly relaxed by transfer learning [16]. In transfer learning, the backbone is first pre-trained to perform some task involving a very large training dataset, e.g., performing image classification on ImageNet [17]. Visual concepts learned by such backbones have been shown to be generally useful and not just applicable to the pre-training task [18]. The backbone is then fine-tuned on the target task, which usually involves a much smaller dataset. Backbones are usually only pre-trained on 2D RGB images, as these are ubiquitous. However, recently, *Omnivore* [19] was introduced. *Omnivore* is a method to train a shifted window transformer [20] backbone on any combination of congruent color and depth images. We therefore used *Omnivore* to serve as a depth-aware backbone in our instance segmentation model.

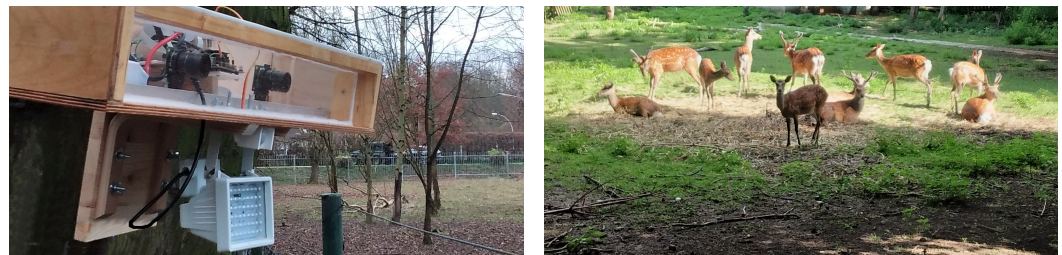
## 2.3. Abundance Estimation

There exist a number of methods to estimate the density and abundance of unmarked animal populations using camera traps: the random encounter model (REM) [21], the random encounter and staying time model (REST) [22], the time-to-event model (TTE), the space-to-event model (STE), the instantaneous estimator (IS) [23], and camera trap distance sampling [24]. All of these require an estimation of the effective area surveyed by the camera trap. This area is not simply given by the optical constraints of the camera; instead, it is influenced by factors such as environmental occlusion and the range of the passive infrared sensor, which may not perform consistently at all locations within the camera's viewshed. The effective area surveyed is statistically inferred by using the distances of the observed animals. Although there are approaches that estimate these distances (semi) automatically [25,26], they either require the laborious capture of reference material [25] or might not generalize to extreme scenarios, such as very close-up scenes within 3 m of the camera [26,27].

## 3. Materials and Methods

The SOCRATES camera trap system comprises a cost and power efficient stereovision sensor platform, as well as state-of-the-art animal detection software based on a deep learning software architecture. The experimental evaluation of SOCRATES will utilize a representative dataset generated by SOCRATES in the wildlife park Plittersdorf, located

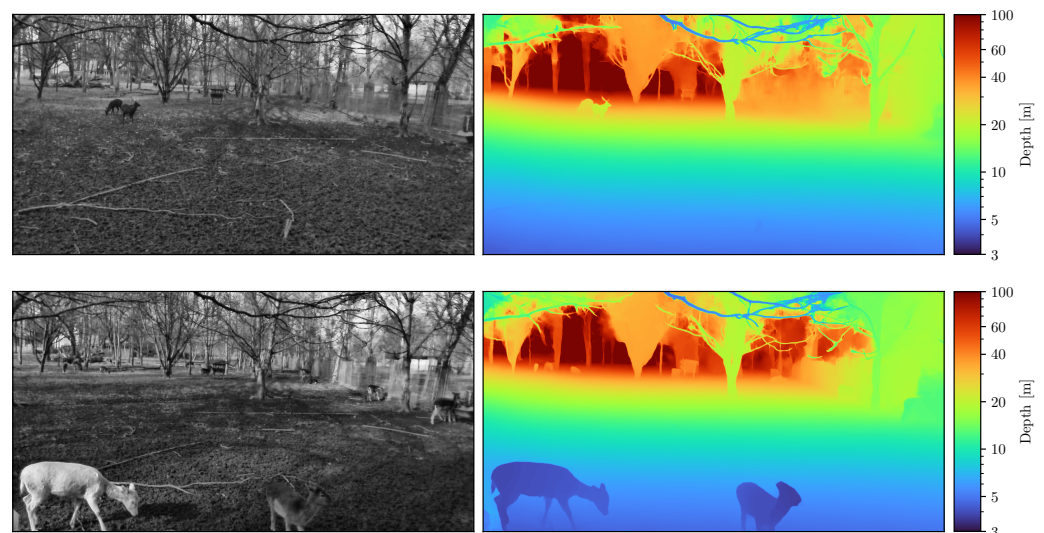
in Bonn, Germany, exhibiting European fallow deer (*Dama dama*) and Sika deer (*Cervus nippon*). Figure 1 illustrates the deployment of SOCRATES in the wildlife park Plittersdorf.



**Figure 1.** Introducing the SOCRATES camera trap system located in the wildlife park Plittersdorf located in Bonn, Germany, showing European fallow deer (*Dama dama*) and Sika deer (*Cervus nippon*).

### 3.1. Data Material

We deployed SOCRATES in the wildlife park Plittersdorf in order to evaluate the hardware and software. Details about this deployment may be found in Section 4. During this time, SOCRATES made 221 true positive observations. Exemplary samples are visualized in Figure 2. Each observation results in an HEVC encoded video with 30 frames per second and a length of 25 s. For our experiments, we sampled two sets of still images from these videos.



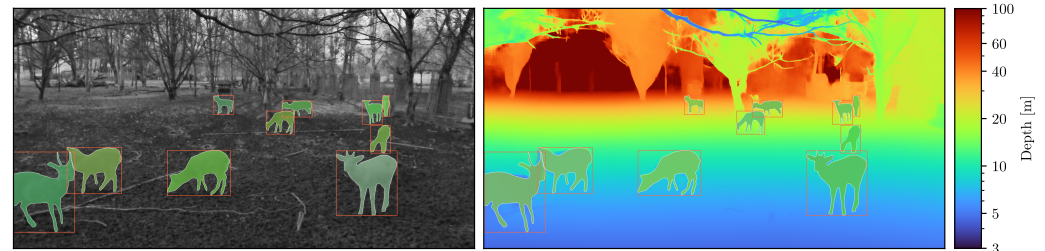
**Figure 2.** Samples of data collected in the wildlife park Plittersdorf. The left shows the grayscale image of the left camera, the right image the color-coded depth map obtained using stereo correspondence.

For our camera trap distance sampling study (see Section 3.4), we sampled still images from the videos at a rate of  $2\text{ s}^{-1}$ , resulting in a total of 2871 images.

For the instance segmentation task (see Section 3.3), we wanted our dataset to consist of diverse scene configurations (animal positions and poses, lighting conditions, etc.). To obtain such diverse samples, sampling at regular intervals is not enough. Sometimes, deer will stand still for long periods of time, while moving quickly through the scene at other times. Therefore, we employed an approach based on background modeling using Gaussian mixture models [28]. We then accumulated the ratio of foreground pixels (that is, the ratio of pixels occupied by moving objects) in each video frame until a threshold of 10% was reached. This way, we sampled more often if there was more movement in the video and less often for less movement. We then annotated a total of 546 instances in 187 of the still images sampled this way with instance masks using the interactive annotation tool proposed by [29]. Figure 3 visualizes one of these annotated images. On average, we needed roughly 3.5 min per instance, resulting in a total annotation effort of roughly



32 h. Out of the total of 546, we used 395 instances for training and validation (via 10-fold cross-validation) and reserved 151 instances as the test dataset, such that images from a single video were only ever contained in one dataset. The test dataset was not used in this work, but was instead reserved for future work. We published both the raw data [30] and the instance segmentation dataset [31].



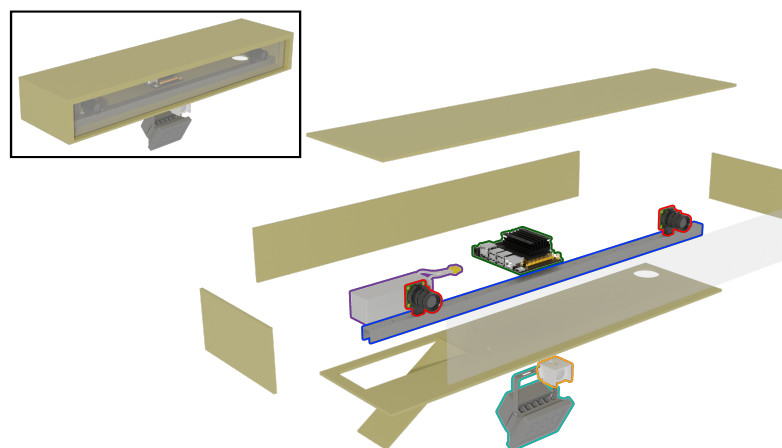
**Figure 3.** Examples of instance segmentation and bounding box annotations overlaid on top of the grayscale image of the left camera (left) and the color-coded depth map (right).

### 3.2. The SOCRATES Stereovision Sensor Platform

The SOCRATES stereovision platform is optimized for:

1. Operability;
  - (a) At day and night time as well as;
  - (b) For a wide range of animal-camera distances;
2. Effective and efficient power supply;
3. Hardware and construction costs;
4. Weather resistance.

Figure 4 illustrates the final hardware design, while the following section covers in detail the technical implementation and how we addressed these design goals. We first describe the stereo camera design (cameras and baseline, Design Goals 1 and 3). The raw data produced by the cameras were processed and stored by the control unit (Design Goals 2 and 3). Weather resistance (Design Goal 4) was provided by the case. Infrared motion detection and illumination facilitate energy efficiency (Design Goal 2) and operability at night time (Design Goal 1 (a)). We additionally describe in detail the power supply, how we obtained animal-camera distances using stereo correspondence, and how the captured data may be transferred using different connectivity options.



**Figure 4.** The 3D visualization of SOCRATES. The important components will be covered in the following text and are highlighted here, i.e., cameras (red outline), baseline rail (blue outline), control unit (green outline), battery (violet outline), infrared illumination (turquoise outline), passive IR sensor (Orange outline). Details such as the power supply, wiring, and screws are omitted. The following 3D parts of the model were obtained from external sources: Raspberry Pi HQ cameras [32], Jetson Nano Devkit [33], Infrared illuminator [34], PIR sensor case [35], LiPo battery [36].

**Cameras and baseline:** A pair of Raspberry Pi high-quality cameras (Raspberry Pi Foundation, Cambridge, United Kingdom) were chosen for their cost-effectiveness and the high sensitivity of their Sony IMX477 sensor [37]. Interchangeable lenses allow adaptation to specific scenarios (i.e., shorter focal lengths for close-up scenes, higher focal lengths for more distant objects). Removal of the infrared filter allows sufficient exposure at night using artificial infrared illumination, while sacrificing color sensitivity. The cameras were mounted on a 77.5 cm-long U-shaped aluminum rail, which allowed the configuration of different baseline distances between both cameras.

**Control:** An NVIDIA Jetson Nano Developer Kit (NVIDIA Corporation, Santa Clara, United States of America) was used as the central control and storage unit. It is responsible for taking motion detection signals from the PIR sensor, turning on the power to the IR illuminator, and capturing, encoding, and archiving image material from the cameras. The raw RGB video material is encoded on the Jetson Nano's GPU by synchronizing the left and right image streams and encoding the resulting video using the HEVC video codec [38].

**Motion detection:** Like most camera traps, SOCRATES utilizes a pyroelectric infrared (PIR) sensor (HC-SR501, Sertronics GmbH, Berlin, Germany) for detecting motion and, thereby, triggering capture.

**Illumination:** A 12 W, 850 nm infrared illuminator was employed to ensure properly exposed images at night without disturbing most species.

**Power supply:** All components were powered by a lithium ion polymer battery with a theoretical capacity of 236.8 Wh (16,000 mAh at 14.8 V).

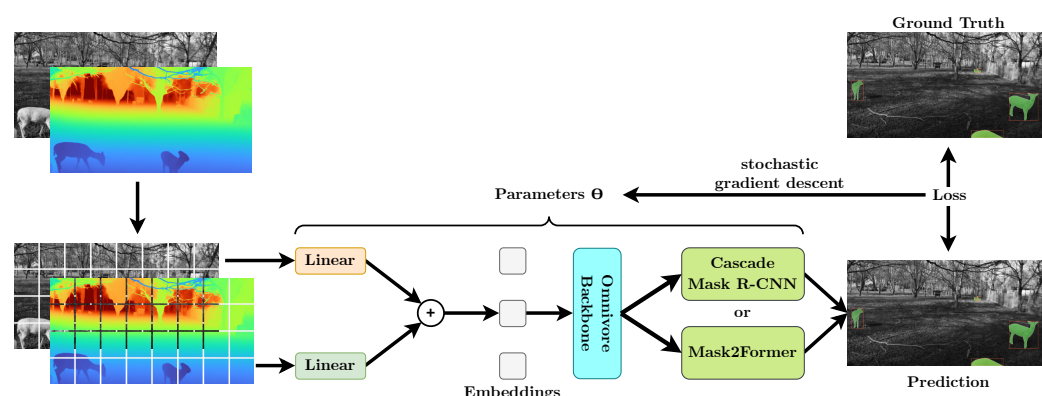
**Connectivity:** SOCRATES may transmit the recorded data to local *AMMOD base stations* [1] via wireless LAN. The AMMOD base stations are able to cache the data and forward them to the web portal of the AMMOD project (see Section 4.5) once conditions are favorable (e.g., abundant energy is available via energy harvesting [2]). If no AMMOD base station is available, data can alternatively be transmitted directly to the AMMOD web portal via a cellular modem.

**Stereo correspondence:** The central goal of SOCRATES is to infer depth information through stereo vision. In the natural world, as well as in computer vision, this is achieved by solving the stereo correspondence problem (see Section 2.1). We performed both intrinsic and extrinsic calibration using *Kalibr* [39] with a grid of  $4 \times 3$  AprilTags [40] mounted on a wooden board as the calibration target. During the setup of SOCRATES, the calibration target was manually moved through the scene such that it covered as much of each camera's field of view as possible. After SOCRATES is assembled and calibrated, calibration does not have to be repeated when deployed to different locations, as the calibration is not dependent on a specific location, but only on the camera configuration. Given the intrinsic and extrinsic parameters, the images of both cameras are rectified, and the disparity of each pixel is computed using [8]. The rectification and disparity computation are performed on a separate GPU server to prolong the battery life of SOCRATES.

### 3.3. Depth-Aware Instance Segmentation

We frame the problem of detecting and localizing animals as an instance segmentation problem, with the goal of generating a bounding box and a binary mask for each animal instance. Compared to animal presence-absence classification, this approach allows both counting the exact number of animals present, as well as inferring the distance between animal and camera by applying the binary mask to the depth images obtained using stereo vision. However, the depth images themselves obtain useful information for differentiating multiple individual animals from themselves and the background. Still, it is not obvious how to use the depth images obtained from SOCRATES in this framework. Compared to datasets such as ShapeNet [41], we only have information from (effectively) a single perspective. We therefore argue that it is wise to treat the depth information as an additional channel in the two-dimensional image instead of working on point clouds or voxel grids, which increase the computational and memory requirements while largely foregoing the significant improvements being made in the area of 2D instance segmentation. Although

most backbones are pre-trained on color images without depth information, a recent work proposed *Omnivore*, a vision transformer backbone trained on color and depth information [19]. In our experiments, we used *Omnivore* as our backbone of choice. As instance segmentation models, we used either the convolution-based *Cascade Mask R-CNN* [42] or the transformer-based *Mask2Former* [15]. This is motivated by the observation that vision transformers require more training data to perform well, compared to CNNs [43,44]. Therefore, *Mask2Former* performs very poorly when trained on small datasets such as the *Plittersdorf instance segmentation dataset*, while outperforming *Cascade Mask R-CNN* on larger datasets such as *Cityscapes* [45]. The resulting model architecture is visualized by Figure 5. To demonstrate that depth information is not only beneficial on the *Plittersdorf instance segmentation task*, we also evaluated improvements on the *Cityscapes instance segmentation dataset*. This is because the *Cityscapes dataset* is one of the only datasets that provides both depth information through stereo vision and a large amount of instance segmentation annotations. We implemented our instance segmentation pipeline using *mmdetection* [46] and largely kept the default hyperparameters of the *mmdetection* model implementations, including basic color, horizontal flip, and random cropping augmentations. We used the AdamW optimizer [47] with a global learning rate of  $5N10^{-5}$  with a batch size of  $N$  and a weight decay of 0.05. We set  $N = 2$  for *Mask2Former* and  $N = 6$  for *Cascade Mask R-CNN* due to memory constraints. To conserve the battery life of SOCRATES, we ran the instance segmentation model on a dedicated GPU-equipped server (see Section 4.5).



**Figure 5.** The data flow of our RGB-D instance segmentation model using *Omnivore* [19] and *Cascade Mask R-CNN* [42] or *Mask2Former* [15]. In *Omnivore*, the grayscale and depth images are first split into 2D patches, linearly embedded and added together. The resulting embeddings are then passed through the *Omnivore* backbone, which generates hierarchical feature maps. These feature maps are then used by *Cascade Mask R-CNN* or *Mask2Former* to perform instance segmentation. The entire model is optimized via stochastic gradient descent via the AdamW optimizer [47]. The model architectures and loss computation are described in detail in [15,42].

### 3.4. Camera Trap Distance Sampling Study

It is now possible to combine the instance masks generated by the animal detection model (see Section 3.3) with the depth images obtained by stereo vision (see Section 2.1) to obtain the distances required for the camera trap distance sampling (CTDS) [24] abundance estimation method. To be able to use all observations without leaking information from the training dataset of our instance segmentation model, in this study, we used not the instance masks, but the bounding boxes of MegaDetector [48] and the sampling approach by [25]. To show the viability of this approach, we performed an exemplary estimation of the detection probability. We used 7 equally spaced distance intervals from 3 m to 11 m. As SOCRATES is mounted on a tree just outside the enclosure and at a height of 1.9 m, 3 m is the minimal distance where deer are certain to be visible. We did not re-scale the minimum distance, as deer might be present closer to the camera, but outside the field

of view. We used the *Distance for Windows* software (version 7.4 [49]) and modeled the detection function using a uniform key function with a single cosine adjustment term.

#### 4. Evaluation

We operated SOCRATES in the wildlife park Plittersdorf, Bonn, Germany, from February 9th to July 8th 2022, or 149 days. The wildlife park Plittersdorf houses exclusively European fallow deer (*Dama dama*) and Sika deer (*Cervus nippon*). The camera was mounted on the side of a tree using a lashing strap and not moved during the entire duration. During this time, SOCRATES experienced temperatures from  $-4^{\circ}\text{C}$  to  $38^{\circ}\text{C}$  and storms with wind speeds of  $87\text{ km h}^{-1}$  without issues. SOCRATES was without power or the software disabled due to maintenance for 46 days, resulting in a total number of 103 observation days. During this time, SOCRATES recorded 1089 observations. Out of these, 221 showed visible animals. This indicates a false positive rate of roughly 80%, which is in line with prior work concerned with commercial camera traps [50]. False triggers are primarily induced by (1) animals in the field of view of the PIR sensor, but outside of the fields of view of the cameras, (2) excessive infrared illumination by the Sun during daytime, or (3) by artificial light sources such as flood lights on nearby buildings. We manually removed all false positive observations from our dataset. Although this could easily be automated, e.g., by using the MegaDetector [48], we wanted to ensure that there were no persons in the final dataset and, therefore, screened the entire dataset manually.

We compare SOCRATES with the widely used commercially produced Reconyx HP2XC in Table 1. SOCRATES is significantly larger to support large baselines, while having significantly shorter battery life and slightly higher component costs. The infrared illuminator of SOCRATES operates at a slightly shorter wavelength, which might be visible for some animals. The infrared illuminator should therefore be replaced with a longer-wavelength version in the future. At the same time, SOCRATES not only provides depth information through stereo vision, but also allows recording video at high resolutions and frame rates for long durations, only limited by available storage space. The interchangeable lenses, as well as the configurable baseline construction allow for adaptation to specific scenarios, e.g., free fields, feeding places, animal crosses, green bridges, etc., where animals are observed at different distances.

**Table 1.** Comparison between SOCRATES and the widely used Reconyx HP2XC trail camera. The video length of SOCRATES is only bounded by the available persistent storage space, although we used 25 s videos for our experiments. Costs include only materials and not assembly.

|                         | HP2XC [51]                            | SOCRATES                             |
|-------------------------|---------------------------------------|--------------------------------------|
| Provided Depth          | ✗                                     | ✓                                    |
| Image Resolution        | $1920 \times 1080/3\text{MP}$         | $1920 \times 1080$                   |
| Video Resolution        | $1280 \times 720$                     | $1920 \times 1080$                   |
| Video Length            | max. 90 s at 2FPS                     | Up to 40 h at 30FPS                  |
| Daytime Imaging         | RGB                                   | Near-infrared                        |
| Nighttime Imaging       | Near-infrared                         | Near-infrared                        |
| Illumination Wavelength | 940 nm                                | 850 nm                               |
| Dimensions              | $14 \times 11.5 \times 7.5\text{ cm}$ | $11.6 \times 80 \times 20\text{ cm}$ |
| Connectivity            | Cellular                              | Cellular/W-LAN/LAN                   |
| Battery Life            | Up to a year                          | ~9 days                              |
| Material Cost           | USD 659.99                            | ~USD 900                             |

We demonstrate that the stereo capabilities SOCRATES facilitate improved visual animal detection (see Section 4.2) and accurate abundance estimation using camera trap distance sampling (see Section 4.4). Depth information is also essential for obtaining absolute animal sizes (referenced as photogrammetry in ecology [52]), which is traditionally performed using laser rangefinders. Furthermore, depth information has been shown to improve the accuracy of animal tracking over 2D-only approaches [53,54]. SOCRATES



cannot compete with commercially available camera traps in cost or battery life, but this was not our goal. Apart from the methodological improvements described above, SOCRATES fulfills three high-level goals:

1. It demonstrates that stereo camera traps are viable and worthwhile. We hope to convince commercial camera trap manufacturers to support stereo camera setups using off-the-shelf hardware.
2. It facilitates the verification of monocular approaches. For example, abundance estimation using camera trap distance sampling might be performed twice, once using monocular approaches [25,26] and once using SOCRATES. Both raw animal distances and the resulting animal densities might then be compared.
3. It allows generating training data for monocular depth estimation methods such as [55–57]. These approaches have been largely focused on human-centric scenes such as indoor and street scenes with a relatively simple geometry, which are highly unlike natural scenes such as forests. Gathering training data from natural scenes might help these methods generalize better to such scenes and, thus, allow monocular camera traps to more accurately estimate depth information in the future.

#### 4.1. Stereo Correspondence

We evaluated the accuracy of SOCRATES quantitatively by comparing the depth obtained from stereo correspondence (see Section 2.1) with  $n = 6$  pointwise ground truth measurements  $Y$  obtained using a laser rangefinder. For the ground truth measurements, we picked distinct objects such as tree trunks to be able to accurately re-identify them in the SOCRATES images. As metrics, we employed the root-mean-squared (RMSE) and end-to-end point (EPE) errors in disparity space. We obtained a root-mean-squared error of 1.63px and an EPE of 1.43px, which are in line with prior work on comparable scenes [5,6].

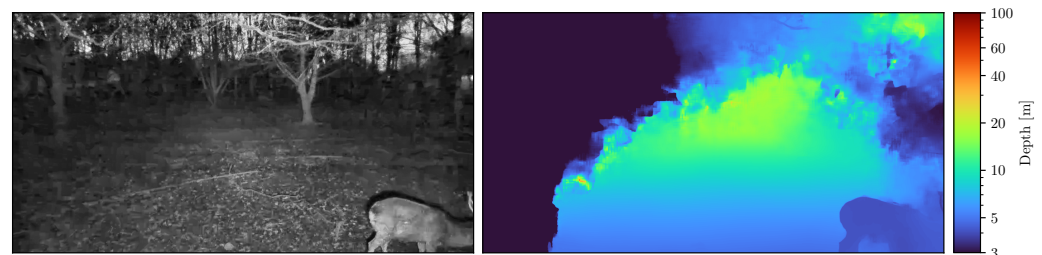
We additionally evaluated the temporal stability of the depth maps using the temporal quality metric proposed in [58], which is defined as:

$$E_t = \frac{1}{(N_T - 1)N_P} \sum_{n=2}^{N_T} \sum_{(x,y)} |D(x, y, n) - D(x - m_x, y - m_y, n - 1)| \quad (3)$$

where  $N_T$  is equal to the number of frames in the input video,  $N_P$  is the number of pixels in a single frame,  $D(x, y, n)$  is the scalar disparity at some pixel  $(x, y)$  at time  $n$ , and  $m_x, m_y$  is the optical flow from frame  $n$  to frame  $n - 1$ , calculated using [59]. Using [8], we obtained  $E_t = 0.4439$ , which is on par with the temporal error of the ground truth disparity in [58].

Figure 2 shows some exemplary pairs of near-infrared images and corresponding depth maps inferred by [8]. As can be seen, the depth maps generally represent the scene well and clearly highlight the boundaries of the deer.

Like regular camera traps, at night time, some regions in the field of view might be insufficiently lit and, therefore, underexposed in the resulting images. In these regions, insufficient image information is available to perform successful stereo correspondence, which is illustrated by Figure 6.



**Figure 6.** Stereo matching inevitably fails in regions where there is not enough available information.

#### 4.2. Visual Animal Detection

We used the COCO [60] metrics to evaluate our instance segmentation models. Each metric was obtained by performing 10-fold cross-validation after the last training epoch. Cross-validation is especially important in this setting, as it reduces the impact of a single lucky train–test split on this small dataset. Table 2 summarizes the results on the Plittersdorf instance segmentation task. The summarizing metrics for bounding boxes ( $AP^{bbox}$ ) and segmentation ( $AP^{segm}$ ) show that incorporating depth information results in an overall performance improvement. Interestingly, for low IOU thresholds ( $AP_{50}^{bbox}$ ,  $AP_{50}^{segm}$ ), depth information seems to have the opposite effect. In other words, pure grayscale images perform better for roughly localizing an animal, whereas grayscale and depth information together are better for localizing animals very accurately ( $AP_{75}^{bbox}$ ,  $AP_{75}^{segm}$ ). This is especially interesting as the ground truth labeling was performed using exclusively the grayscale image. Intuitively, one could therefore argue that the grayscale information is most important for matching the ground truth very precisely. Here, we see the opposite effect. As the error of stereo correspondence is quadratically related to the true distance, the resulting depth maps become less useful at larger distances. This is reflected in the lower performance on small instances ( $AP_s^{segm}$ ), which are typically farther away than medium ( $AP_m^{segm}$ ) or large instances ( $AP_l^{segm}$ ). We tried to ease the dependence on depth information for these faraway instances by clipping the depth values to different maximum distances or randomly dropping the depth information altogether during training [61]. However, this did not result in meaningful improvements. We provide benchmarks of inference and training time in Table A2 and an experimental comparison with more lightweight ResNet backbones in Table A1.

**Table 2.** COCO metrics on the Plittersdorf instance segmentation dataset task using Cascade Mask R-CNN with different backbones and 10-fold cross-validation. The respective best value for each metric is highlighted in bold.

| Backbone          | $AP^{bbox}$      | $AP_{75}^{bbox}$ | $AP^{segm}$   | $AP_{50}^{segm}$ |
|-------------------|------------------|------------------|---------------|------------------|
| Swin-L            | 0.5164           | 0.5272           | 0.4359        | 0.8328           |
| Omnivore-L        | 0.5243           | 0.5702           | 0.4382        | <b>0.8353</b>    |
| + depth-awareness | <b>0.5399</b>    | <b>0.6048</b>    | <b>0.4547</b> | 0.8147           |
| Backbone          | $AP_{75}^{segm}$ | $AP_l^{segm}$    | $AP_m^{segm}$ | $AP_s^{segm}$    |
| Swin-L            | 0.4285           | 0.5192           | 0.3856        | 0.1285           |
| Omnivore-L        | 0.4376           | 0.515            | 0.3895        | <b>0.1431</b>    |
| + depth-awareness | <b>0.4699</b>    | <b>0.5427</b>    | <b>0.4013</b> | 0.1138           |

#### 4.3. Depth-Aware Instance Segmentation on Cityscapes

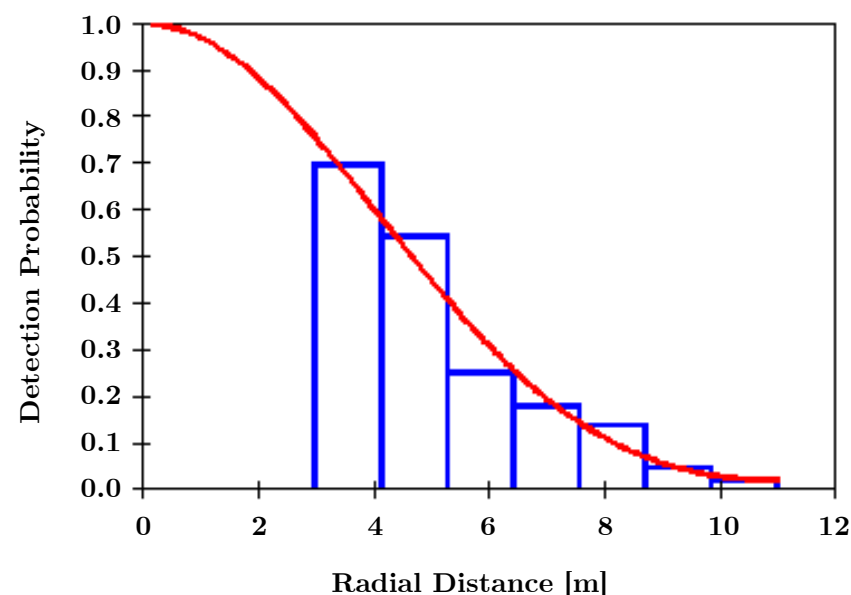
To show that the positive effect of depth information on instance segmentation accuracy is not limited to settings with grayscale images, a single object class, and a fixed camera such as SOCRATES, we additionally evaluate our instance segmentation approach on the Cityscapes instance segmentation task [45]. The Cityscapes instance segmentation dataset [45] is composed of color and stereo depth images of urban street scenes, captured by cameras in a moving car. It features several object classes, such as *person*, *car*, or *bus*, annotated with instance labels. The Cityscapes dataset is also much larger, with 3475 annotated images in its training and validation sets. As can be seen in Table 3, the depth information has an overall even greater positive impact than in the Plittersdorf task (see Section 4.2). This is likely caused by two reasons: (1) the Mask2Former [15] being able to better make use of the depth-aware feature hierarchies produced by Omnivore [19] and (2) the larger training dataset, which might help alleviate the lower number of depth images during pre-training [19].

**Table 3.** Instance segmentation results on the Cityscapes validation set. The depth-aware Omnivore-L variant clearly improves the non-depth-aware variant in all metrics. The metrics of the Swin-L backbone were obtained using the original implementation [15].  $AP^{segm}$  and  $AP_{50}^{segm}$  are Cityscapes metrics; the rest are COCO metrics. The respective best value for each metric is highlighted in bold.

| Backbone               | $AP^{bbox}$  | $AP^{segm}$  | $AP_{50}^{segm}$ | $AP_l^{segm}$ | $AP_m^{segm}$ | $AP_s^{segm}$ |
|------------------------|--------------|--------------|------------------|---------------|---------------|---------------|
| Swin-L                 |              | 0.437        | 0.714            |               |               |               |
| Omnivore-L             | 0.415        | 0.439        | 0.700            | 0.716         | 0.394         | 0.214         |
| Depth-aware Omnivore-L | <b>0.431</b> | <b>0.456</b> | <b>0.734</b>     | <b>0.732</b>  | <b>0.411</b>  | <b>0.264</b>  |

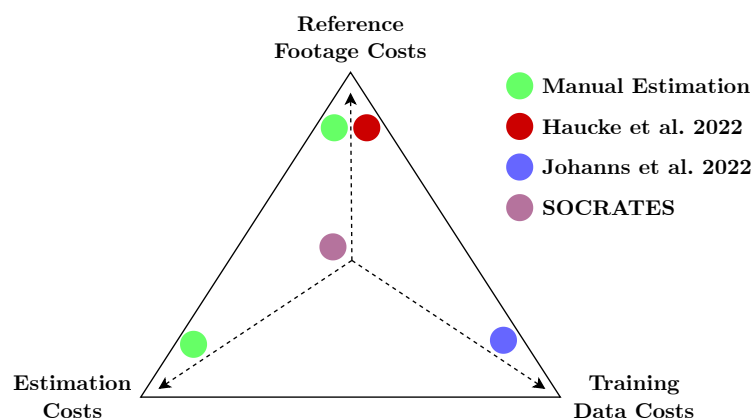
#### 4.4. Abundance Estimation Using SOCRATES

Figure 7 depicts the detection probability obtained by CTDS using the parameters specified in Section 3.4. Note that the estimated probability density approximates the measurements well, starting from a distance of 3 m. Due to the way SOCRATES is mounted, deer below 3 m may not be visible, which is why we excluded these low distances from our estimation (see Section 3.4).



**Figure 7.** CTDS detection probability. We transformed our distance measurements into seven intervals (visualized in blue) from which the detection probability (visualized in red) was derived using CTDS.

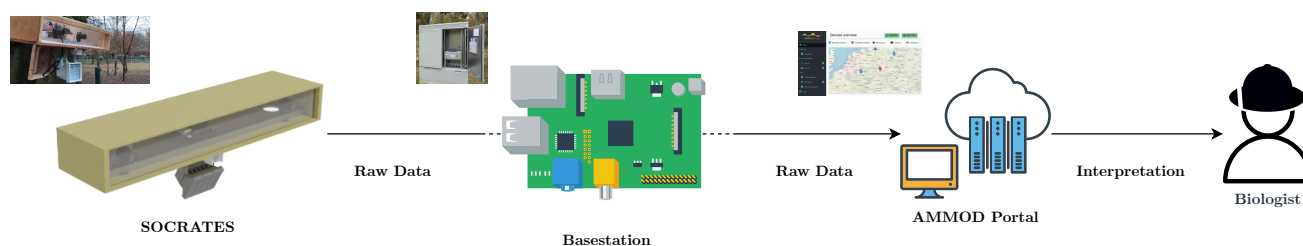
Compared to competing approaches [25,26], distance estimation for abundance estimation of unmarked animal populations is straightforward with SOCRATES. Figure 8 visualizes the respective tradeoffs. The presented proof-of-concept for modeling detection probability in camera trap distance sampling with SOCRATES demonstrates the superiority of using SOCRATES in CTDS by significantly improving the efficacy of future wildlife surveys [25]. The reduced cost for data processing, the increase in animal detection, and potential for application in integrated mono- and stereo-camera trap surveys pave the way for an end-to-end solution in computational wildlife monitoring. The proposed approach is not limited to the conditions of our study, but is widely applicable across habitats, species, and regions. For future field surveys, we recommend that multiple SOCRATES devices be used, along with a random or systematic study design to estimate wildlife density and associated variance reliably. SOCRATES can also be paired cooperatively with traditional, monocular camera traps for improved error quantification and improvement of monocular distance estimations like that proposed by [26].



**Figure 8.** Tradeoff between reference footage costs, manual distance estimation costs, and training data costs. The manual approach requires both the acquisition of reference footage and labor-intensive manual animal distance estimation. Reference [25] requires reference footage, but the distance estimation itself is automated. Reference [26] requires neither reference footage nor manual distance estimations, but requires training data of similar scenes. SOCRATES measures distance using stereo vision and, therefore, incurs none of these costs.

#### 4.5. AMMOD Portal Case Study

A central goal of the AMMOD project is to automatically collect all observed data in a central repository (the *AMMOD Portal*, <https://data.ammmod.de>, accessed on 19 October 2022), which will eventually be accessible to biologists and the general public. For SOCRATES, we ensure this by uploading the captured raw data via the CoAP protocol [62] to the *AMMOD base station* [1,2], if available at the current location, or directly to the AMMOD Portal otherwise. The AMMOD base station takes the role of scheduling and prioritizing data transfer from different sensors according to the energy available from energy harvesting. Once the raw data are uploaded to the AMMOD Portal, a server runs the instance segmentation (see Section 3.3) and distance estimation (see Section 3.4) workflows. To increase throughput and energy efficiency, the server is equipped with an NVIDIA Quadro RTX 6000 GPU to accelerate neural network inference. Both methods are packaged as *Docker* images to simplify dependency management and updates. The resulting instance masks and distances are then again uploaded to the AMMOD Portal and are available for further analysis by biologists. This data flow is fully automated and visualized in Figure 9.



**Figure 9.** Fully automatic flow of data from SOCRATES over the base station to the AMMOD Portal and the expert end users. A GPU server runs the instance segmentation and distance estimation steps and uploads the results back to the AMMOD portal.

## 5. Conclusions

We propose SOCRATES as a novel camera trap approach for automated wildlife monitoring. SOCRATES employs stereo vision to improve animal detection and abundance estimation using depth information. SOCRATES comprises a flexible hardware architecture, as well as a comprehensive deep learning approach for animal detection. The depth information obtained using stereo vision improves the localization accuracy of animals in an instance segmentation setting, for example, by 3.23% in bounding box mAP<sub>75</sub>. The validity of these improvements was underscored by performing 10-fold cross-validation. Similar im-



improvements on the Cityscapes instance segmentation task showed that this effect is neither limited to grayscale images, a single object category, nor fixed cameras such as SOCRATES. The animal localization provided by object detection is combined with the depth information provided by SOCRATES to measure reliable animal observation distances, which are required to estimate animal abundance using methods such as CTDS [24]. SOCRATES achieves this in a fully automated process and independently of large training datasets for monocular depth estimation based on deep learning [26]. We successfully modeled sensible detection probability in a wildlife enclosure using SOCRATES. Future work will use SOCRATES to perform automatic abundance estimation in the wild and compare the results with competing monocular approaches [25,26]. In an effort to make SOCRATES accessible to biologists and ecologists, we openly provide our raw and labeled data, code, detailed instructions, best practices, and 3D CAD models at <https://github.com/timmh/socrates> (accessed on 19 October 2022). We hope to pave the way for the eventual adaption of stereo camera traps by commercial manufacturers.

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

**Funding:** This research was partially funded by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung), Bonn, Germany (AMMOD—Automated Multisensor Stations for Monitoring of BioDiversity: FKZ 01LC1903B).

**Institutional Review Board Statement:** Ethical review and approval were waived for this study due to our experiment in the wildlife park Plittersdorf being non-invasive and employing only passive observation of the animals without disturbance (the camera trap is mounted outside the enclosure and the infrared illumination is invisible to deer). No behavioral experiments were conducted and the deer in the wildlife park Plittersdorf are—independently from our experiment—housed in their natural habitat.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data obtained using SOCRATES are freely available at <https://doi.org/10.5281/zenodo.6992653> (accessed on 20 September 2022) and <https://doi.org/10.5281/zenodo.7035934> (accessed on 20 September 2022).

**Acknowledgments:** We thank Vincent Mainzer and the team of the wildlife park Plittersdorf for their cooperation by hosting the camera trap hardware on site. We thank Frank Schindler for proofreading the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

|            |                                                   |
|------------|---------------------------------------------------|
| SOCRATES   | StereO CameRA Trap for monitoring of biodivErSity |
| CTDS       | camera trap distance sampling                     |
| RGB        | red, green, blue (color channels)                 |
| IR         | infrared                                          |
| PIR sensor | pyroelectric infrared sensor                      |
| HEVC       | High-Efficiency Video Coding                      |

## Appendix A. Comparison with ResNet Backbones

We performed an additional experimental comparison of Omnivore-L [19] and smaller ResNet [63] backbones on the Plittersdorf instance segmentation task. In the depth-aware ResNet variants, we modified the first convolutional layer to accept four instead of three channels, but otherwise used weights pre-trained on ImageNet [17]. The implementation of

the rest of the detection model and the training process remained identical (see Section 3.3). As can be seen in Table A1, the Omnivore-L backbone outperforms the smaller ResNet backbones in all metrics. Still, the ResNet models profit substantially from depth awareness. Additionally, the ResNet backbones are significantly smaller and faster, as illustrated by Table A2. This potentially makes running the ResNet-based instance segmentation model directly on SOCRATES more feasible.

**Table A1.** COCO metrics on the Plittersdorf instance segmentation dataset task using Cascade Mask R-CNN with depth-aware Omnivore-L and ResNet backbones, evaluated using 10-fold cross-validation. The respective best value for each metric is highlighted in bold.

| Backbone               | AP <sup>bbox</sup>               | AP <sup>bbox</sup> <sub>75</sub> | AP <sup>segm</sup>              | AP <sup>segm</sup> <sub>50</sub> |
|------------------------|----------------------------------|----------------------------------|---------------------------------|----------------------------------|
| Depth-aware Omnivore-L | <b>0.5399</b>                    | <b>0.6048</b>                    | <b>0.4547</b>                   | <b>0.8147</b>                    |
| ResNet-18              | 0.3361                           | 0.2754                           | 0.2788                          | 0.6669                           |
| + depth-awareness      | 0.3318                           | 0.3194                           | 0.3082                          | 0.6408                           |
| ResNet-50              | 0.3744                           | 0.3624                           | 0.3038                          | 0.6936                           |
| + depth-awareness      | 0.4122                           | 0.3804                           | 0.3486                          | 0.7164                           |
| Backbone               | AP <sup>segm</sup> <sub>75</sub> | AP <sup>segm</sup> <sub>l</sub>  | AP <sup>segm</sup> <sub>m</sub> | AP <sup>segm</sup> <sub>s</sub>  |
| Depth-aware Omnivore-L | <b>0.4699</b>                    | <b>0.5427</b>                    | <b>0.4013</b>                   | <b>0.1138</b>                    |
| ResNet-18              | 0.1585                           | 0.3129                           | 0.2807                          | 0.0987                           |
| + depth-awareness      | 0.2433                           | 0.3487                           | 0.2615                          | 0.0563                           |
| ResNet-50              | 0.2227                           | 0.3572                           | 0.3108                          | 0.0703                           |
| + depth-awareness      | 0.2475                           | 0.4128                           | 0.3009                          | 0.0724                           |

**Table A2.** Benchmarks of the two ResNet [63] and the Omnivore-L [19] backbones on a single NVIDIA Quadro RTX 6000.

| Backbone   | Parameters | Training Duration (h) | Inference Time Per Image (ms) |
|------------|------------|-----------------------|-------------------------------|
| ResNet-18  | 11M        | 3.5                   | 35.0                          |
| ResNet-50  | 24M        | 5.4                   | 102.2                         |
| Omnivore-L | 196M       | 28.1                  | 471.1                         |

## References

1. Wägele, J.; Bodesheim, P.; Bourlat, S.J.; Denzler, J.; Diepenbroek, M.; Fonseca, V.; Frommolt, K.H.; Geiger, M.F.; Gemeinholzer, B.; Glöckner, F.O.; et al. Towards a multisensor station for automated biodiversity monitoring. *Basic Appl. Ecol.* **2022**, *59*, 105–138. [\[CrossRef\]](#)
2. Sixdenier, P.L.; Wildermann, S.; Ziegler, D.; Teich, J. SIDAM: A Design Space Exploration Framework for Multi-sensor Embedded Systems Powered by Energy Harvesting. In Proceedings of the Embedded Computer Systems: Architectures, Modeling, and Simulation, Samos, Greece, 3–7 July 2022; Orailoglu, A., Reichenbach, M., Jung, M., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 329–345.
3. Longuet-Higgins, H.C. A computer algorithm for reconstructing a scene from two projections. *Nature* **1981**, *293*, 133–135. [\[CrossRef\]](#)
4. Zbontar, J.; LeCun, Y. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. *J. Mach. Learn. Res.* **2016**, *17*, 65:1–65:32.
5. Chang, J.R.; Chen, Y.S. Pyramid Stereo Matching Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 15–23 June 2018; pp. 5410–5418.
6. Zhang, F.; Prisacariu, V.; Yang, R.; Torr, P.H. GA-Net: Guided Aggregation Net for End-to-end Stereo Matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 185–194.
7. Xu, H.; Zhang, J. AANet: Adaptive Aggregation Network for Efficient Stereo Matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1959–1968.
8. Li, J.; Wang, P.; Xiong, P.; Cai, T.; Yan, Z.; Yang, L.; Liu, J.; Fan, H.; Liu, S. Practical stereo matching via cascaded recurrent network with adaptive correlation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022; pp. 16263–16272.
9. Xu, Z.; Sun, L.; Wang, X.; Dong, H.; Lei, P.; Zhou, Y. Binocular camera trap for wildlife detection. In Proceedings of the Optoelectronic Imaging and Multimedia Technology VI, Hangzhou, China, 21–23 October 2019; Dai, Q., Shimura, T., Zheng, Z., Eds.; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA, 2019; Volume 11187, p. 111871C. [\[CrossRef\]](#)

10. Xu, Z.; Sun, L.; Wang, X.; Lei, P.; He, J.; Zhou, Y. Stereo camera trap for wildlife in situ observations and measurements. *Appl. Opt.* **2020**, *59*, 3262–3269. [CrossRef]
11. Haucke, T.; Steinhage, V. Exploiting Depth Information for Wildlife Monitoring. *arXiv* **2021**, arXiv:cs.CV/2102.05607.
12. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *arXiv* **2019**, arXiv:1905.05055.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]
14. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. *arXiv* **2017**, arXiv:1703.06870.
15. Cheng, B.; Misra, I.; Schwing, A.G.; Kirillov, A.; Girdhar, R. Masked-attention Mask Transformer for Universal Image Segmentation. *arXiv* **2022**, arXiv:2112.01527.
16. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In Proceedings of the International Conference on Artificial Neural Networks, Shenzhen, China, 8–10 December 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 270–279.
17. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.
18. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.
19. Girdhar, R.; Singh, M.; Ravi, N.; van der Maaten, L.; Joulin, A.; Misra, I. Omnivore: A Single Model for Many Visual Modalities. *arXiv* **2022**, arXiv:2201.08377.
20. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021.
21. Rowcliffe, J.M.; Field, J.; Turvey, S.T.; Carbone, C. Estimating animal density using camera traps without the need for individual recognition. *J. Appl. Ecol.* **2008**, *45*, 1228–1236. [CrossRef]
22. Nakashima, Y.; Fukasawa, K.; Samejima, H. Estimating animal density without individual recognition using information derivable exclusively from camera traps. *J. Appl. Ecol.* **2018**, *55*, 735–744. [CrossRef]
23. Moeller, A.K.; Lukacs, P.M.; Horne, J.S. Three novel methods to estimate abundance of unmarked animals using remote cameras. *Ecosphere* **2018**, *9*, e02331. [CrossRef]
24. Howe, E.J.; Buckland, S.T.; Després-Einspenner, M.L.; Kühl, H.S. Distance sampling with camera traps. *Methods Ecol. Evol.* **2017**, *8*, 1558–1565. [CrossRef]
25. Haucke, T.; Kühl, H.S.; Hoyer, J.; Steinhage, V. Overcoming the distance estimation bottleneck in estimating animal abundance with camera traps. *Ecol. Inform.* **2022**, *68*, 101536 [CrossRef]
26. Johanns, P.; Haucke, T.; Steinhage, V. Automated distance estimation for wildlife camera trapping. *Ecol. Inform.* **2022**, *70*, 101734. doi: 10.1016/j.ecoinf.2022.101734. [CrossRef]
27. Auda, E. (Wildlife Conservation Society, Phnom Penh, Cambodia). Overestimation of Animal Distances in Close-Up Scenarios. Personal communication, 2022
28. KaewTraKulPong, P.; Bowden, R. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-Based Surveillance Systems*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 135–144.
29. Sofiiuk, K.; Petrov, I.A.; Konushin, A. Reviving iterative training with mask guidance for interactive segmentation. *arXiv* **2021**, arXiv:2102.06583
30. Haucke, T.; Steinhage, V. SOCRATES Plittersdorf Raw Dataset. 2022. Available online: <https://zenodo.org/record/6992653#.Y325eX1BxPY> (accessed on 20 September 2022).
31. Haucke, T.; Steinhage, V. SOCRATES Plittersdorf Instance Segmentation Dataset. 2022. Available online: <https://zenodo.org/record/7035934#.Y325c31BxPY> (accessed on 20 September 2022).
32. Rasool, I. Raspberry Pi HQ Camera with arduCAM CS Mount Lens. Available online: <https://grabcad.com/library/raspberry-pi-hq-camera-with-arducam-cs-mount-lens-1> (accessed on 16 July 2022).
33. Minichiello, S. nVidia Jetson Nano Development Board. Available online: <https://grabcad.com/library/nvidia-jetson-nano-development-board-1> (accessed on 16 July 2022).
34. Medina, J.A.V. Infrared Illuminator. Available online: <https://grabcad.com/library/infrared-illuminator-1> (accessed on 16 July 2022).
35. Machado, M. PIR Sensor Wall Mount Enclosure. Licensed under the Creative Commons Attribution 4.0 International License. Available online: <https://www.thingiverse.com/thing:1718985> (accessed on 16 July 2022).
36. Stoudek, P. Tattu 4S 6750mAh LiPo Battery. Available online: <https://grabcad.com/library/tattu-4s-6750mah-lipo-battery-1> (accessed on 16 July 2022).
37. Sony Semiconductor Solutions Corporation. IMX477-AACK Product Information. Available online: [https://www.sony-semicon.co.jp/products/common/pdf/IMX477-AACK\\_Flyer.pdf](https://www.sony-semicon.co.jp/products/common/pdf/IMX477-AACK_Flyer.pdf) (accessed on 16 July 2022).
38. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [CrossRef]
39. Maye, J.; Furgale, P.; Siegwart, R. Self-supervised calibration for robotic systems. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, 23 June 2013; pp. 473–480.

40. Olson, E. AprilTag: A robust and flexible visual fiducial system. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3400–3407.
41. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012.
42. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
43. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
44. Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; Shi, H. Escaping the Big Data Paradigm with Compact Transformers. *arXiv* **2021**, arXiv:2104.05704.
45. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
46. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
47. Loshchilov, I.; Hutter, F. Fixing Weight Decay Regularization in Adam. *arXiv* **2017**, arXiv:1711.05101.
48. Beery, S.; Morris, D.; Yang, S. Efficient Pipeline for Camera Trap Image Review. *arXiv* **2019**, arXiv:1907.06772.
49. Thomas, L.; Buckland, S.T.; Rexstad, E.A.; Laake, J.L.; Strindberg, S.; Hedley, S.L.; Bishop, J.R.; Marques, T.A.; Burnham, K.P. Distance software: Design and analysis of distance sampling surveys for estimating population size. *J. Appl. Ecol.* **2010**, *47*, 5–14. [[CrossRef](#)]
50. Newey, S.; Davidson, P.; Nazir, S.; Fairhurst, G.; Verdicchio, F.; Irvine, R.J.; van der Wal, R. Limitations of recreational camera traps for wildlife management and conservation research: A practitioner’s perspective. *Ambio* **2015**, *44*, 624–635. [[CrossRef](#)]
51. Reconyx. HP2XC HyperFire 2 Cellular Professional Covert IR Camera. Available online: <https://www.reconyx.com/product/hyperfire-2-cellular-professional-covert-ir-camera> (accessed on 22 June 2022).
52. Shrader, A.M.; Ferreira, S.M.; Van Aarde, R.J. Digital photogrammetry and laser rangefinder techniques to measure African elephants. *S. Afr. J. Wildl. Res.* **2006**, *36*, 1–7.
53. Klasen, M.; Steinhage, V. Improving wildlife tracking using 3D information. *Ecol. Inform.* **2022**, *68*, 101535. [[CrossRef](#)]
54. Klasen, M.; Steinhage, V. Wildlife 3D multi-object tracking. *Ecol. Inform.* **2022**, *71*, 101790. [[CrossRef](#)]
55. Godard, C.; Mac Aodha, O.; Firman, M.; Brostow, G.J. Digging into Self-Supervised Monocular Depth Prediction. *arXiv* **2019**, arXiv:1806.01260.
56. Ranftl, R.; Lasinger, K.; Hafner, D.; Schindler, K.; Koltun, V. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1623–1637. [[CrossRef](#)]
57. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision Transformers for Dense Prediction. *arXiv* **2021**, arXiv:2103.13413.
58. Vandewalle, P.; Varekamp, C. Disparity map quality for image-based rendering based on multiple metrics. In Proceedings of the 2014 International Conference on 3D Imaging (IC3D), Liège, Belgium, 9–10 December 2014; pp. 1–5.
59. Farnebäck, G. Two-frame motion estimation based on polynomial expansion. In Proceedings of the Scandinavian Conference on Image Analysis, Halmstad, Sweden, 29 June–2 July 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 363–370.
60. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, arXiv:1405.0312.
61. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
62. Bormann, C.; Castellani, A.P.; Shelby, Z. CoAP: An Application Protocol for Billions of Tiny Internet Nodes. *IEEE Internet Comput.* **2012**, *16*, 62–67. [[CrossRef](#)]
63. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.