DeepWILD : Wildlife Identification, Localisation and estimation on camera trap videos using Deep learning

Fanny Simões, Charles Bouveyron, Frédéric Precioso

DeepWILD: Wildlife Identification, Localisation and estimation on camera trap videos using Deep learning

Fanny Simões[a], Charles Bouveyron[b], Frédéric Precioso[b]

[a] *Université Côte d'Azur, Institut 3IA Côte d'Azur, 3IATechPool, France*
[b] *Université Côte d'Azur, INRIA, CNRS, Maasai team, France*

Abstract

Videos and images from camera traps are more and more used by ecologists to estimate the population of species on a territory. It is a laborious work since experts have to analyse massive data sets manually. This takes also a lot of time to filter these videos when many of them do not contain animals or are with human presence. Fortunately, deep learning algorithms for object detection can help ecologists to identify multiple relevant species on their data and to estimate their population. In this study, we propose to go even further by using object detection model to detect, classify and count species on camera traps videos. To this end, we developed a 3-step process: (i) At the first stage, after splitting videos into images, we annotate images by associating bounding boxes to each label thanks to MegaDetector algorithm; (ii) then, we extend MegaDetector based on Faster R-CNN architecture with backbone Inception-ResNet-v2 in order to not only detect the 13 relevant classes but also to classify them; (iii) finally, we design a method to count individuals based on the maximum number of bounding boxes detected. This final stage of counting is evaluated in two different contexts: first including only detection results (i.e. comparing our predictions against

the right number of individuals, no matter their true class), then an evolved version including both detection and classification results (i.e. comparing our predictions against the right number in the right class). The results obtained during the evaluation of our model on the test data set are: (i) 73,92% mAP for classification, (ii) 96,88% mAP for detection with a ratio Intersection-Over-Union (IoU) of 0.5 (overlapping ratio between groundtruth bounding box and the detected one), and (iii) 89,24% mAP for detection at IoU=0.75. Highly represented classes, like humans, have highest values of mAP around 81% whereas less represented classes in the train data set, such as dogs, have lowest values of mAP around 66%. Regarding the proposed counting method, we predicted a count either exact or $\pm$ 1 unit for 87% with detection results and for 48% with detection and classification results of our test data set. Our model is also able to detect empty videos. To the best of our knowledge, this is the first study in France about the use of object detection model on a French national park to locate, identify and estimate the population of species from camera trap videos.
Keywords: Camera trap, CNN, Deep learning, Image classification, Object detection

1. Introduction

The potential of Machine Learning (ML) on wildlife conservation has been more and more investigated in the last years, to address diverse tasks such as recognizing species from different modalities (audio for birds or cetaceans, visual for animals or humans, ...), tracking and pose estimation, etc. This potential has been well described in Tuia et al. (2022).

Among all modalities and sensors explored in the recent works, camera traps are increasingly exploited to monitor and conserve species, with professional users such as researchers in Ecology, or private individuals who want to detect and track animals on their own property. For species preservation, ecologists need to identify species presence on a territory, estimate their quantity and spatial distribution, to possibly further investigate interactions between species or the impact of the anthropic pressure. In the south of France, the "Parc National du Mercantour" (PNM) aims to monitor the different wildlife species present on its territory in order to better understand how animals live, and thus better protect them, using camera traps. Indeed, this national park located between the Alps and the Mediterranean sea gathers several endangered species. In particular, the wolf has reappeared in the 90's and since it is protected in the park. However, most of ecology researchers who currently use camera traps to monitor wildlife species in the national French parks, have to make this monitoring it manually: they watch each camera trap videos and manually count the different species presence in it. Therefore, the PNM current goal is to develop a tool able to automatically identify and count the different species present on their territory thanks to a network of camera traps. In the future, the PNM would like to wider deploy camera traps that would allow a better spatial and temporal knowledge of the species movements. Deep learning methods appear to be the most suitable options to solve their issue. Thanks to these methods, ecology researchers will be able to estimate the population of common and rare species and they will avoid wasting time on tedious tasks.

In this study, we developed a process to filter empty images extracted

from camera trap videos and to label them thanks to preexisting object detection model and to manual checking. Then, we extended a pretrained model in order to detect relevant classes of interest for the PNM. Finally, we elaborated a method to count individuals on each camera trap video from resulting bounding boxes.

2. Related work

Deep learning. In recent years, deep learning methods are increasingly used for video and frame analyses from camera traps in order to identify and classify species (Wäldchen and Mäder, 2018). It helps ecologists to avoid doing this task manually. Convolutional Neural Network (CNN) is certainly the most used deep learning method for image classification, in other words to identify one or multiple species on images (Chen et al., 2019). Vargas-Felipe et al. (2021) proposed to use not only data from their camera traps but also to augment the training sets with pictures from the web of related species to their study, then they design a pipeline with two possible application scenarios: (i) a binary output targeting the presence or absence of a specific species (in their work they focus on the Desert Bighorn Sheep, DBS) or (ii) a multiclass output aiming 7 species which are often collocated with DBS. In addition to species identification, variants of CNNs can localise species on different frames from a video, it is called object detection (Schneider et al., 2018). For example, as a first step of their classification, (Ferreira et al., 2020) used Mask R-CNN (He et al., 2017), a model for object detection, which automatically localises one of the three studied bird species and crops them in the images. In this work in order to detect and classify the relevant

classes, we fine-tuned MegaDetector based on Faster-RCNN with Inception-ResNet-v2 backbone as object detection model.

Count species. Thanks to object detection methods, which can localise species precisely on images, it is also possible to quantify species on images, which is a key element in the wild life conservation. There are many different approaches to count species on images. The easiest way consists of applying an object detection method and then counting the number of bounding boxes detected by species on each image. We provide a literature review hereafter. For example, Norouzzadeh et al. (2020) simply considered a unique class "animal" then they counted individuals by summing the number of bounding boxes detected on an image with at least 90% of confidence. In order to obtain these bounding boxes, they used a pre-trained object detection model, based on Faster-RCNN object detection algorithm (Ren et al., 2015) and trained their own model on different camera trap data sets. They obtained satisfactory results, since they provided the exact number of animals for 72.4% of images and the predicted count is either exact or $\pm$ 1 unit for 86.8% of images. In addition, in (Beery et al., 2021), authors created a challenge to classify and count species based on bounding boxes detected across camera trap videos. They considered bounding boxes detected with at least 80% of confidence. Thus, for instance, one of their methods is to take the sum of bounding boxes across the sequence as a upper bound of the actual number of individuals. Another method is to take the maximum number of bounding boxes from any image in the sequence as a lower bound of the actual number of individuals across the sequence. The method to count species based on bounding boxes could be also applied on unmanned aerial

vehicle videos. Sarwar et al. (2018) used this technique in order to detect and count sheeps in a paddock to help farmers. For this, they compared two methods, one with R-CNN (Girshick et al., 2014) and another one with hand crafted technique. A similar approach is applied by (Xu et al., 2020) on images captured by a quadcopter, but with another object detection algorithm. In this last work, the authors used pre-trained model Mask R-CNN with a ResNet-101 (He et al., 2016) to detect and count cattle populations.

More accurate methods to count species on a video, like tracking methods, could be used in complementary to object detection algorithms. Currently, these methods are mainly used for counting pedestrians or vehicles rather than counting animals. Nonetheless, it starts to be used in Ecology: Levy et al. (2018) apply the Simple Online Realtime Tracker (SORT) algorithm (Wojke et al., 2017) combined with RetinaNet (Lin et al., 2017) for the detection step in order to detect, classify and count the marine organisms on two different marine video data sets, one evaluation is conducted on aerial video frames and another one on underwater video frames. Another tracking method is used in (Zhang et al., 2018) called Multiple Object Tracking (MOT). It is applied on multiple objects and it estimates the trajectory of each species on frame. It concerns frames from videos of pigs in pens recorded over 3 days by day and night. The combination of object detection and tracking methods consist of testing 3 different CNN detection architectures (Faster-RCNN (Ren et al., 2015), R-FCN (Dai et al., 2016) and SSD (Liu et al., 2016)) with VGG16 (Simonyan and Zisserman, 2014) as backbone and using Discriminative Correlation Filters (DCF) based on on-line tracking method to track each pig.

Among alternative approaches to object detection methods, we can mention (Norouzzadeh et al., 2018) which also calculates the number of species only as a problem of classification, the number of species on images is assigned as a label associated with each image. They used 12 different bins and tested different types of deep neural networks. Notice that it is only used to count one unique species by frame, not multiple species as we consider in this work.

For the counting step in our work, we apply the easiest method based on bounding box detection. On one hand, we count individuals as a whole as in Norouzzadeh et al. (2020) and on the other hand we count only individuals assigned to the relevant classes as in (Beery et al., 2021).

Class imbalance problem. There are multiple challenges associated to study frames of videos from camera traps: blurred, over-exposed, or poor illuminated frames, occlusion, complex animal pose, size of species, daytime or nighttime frames, animals far away from the camera or too close, background variations, multiple species on the same frames, empty images or lack of images. (Villa et al., 2017) enumerate these different issues for species recognition in camera trap image analyses and decided to focus on the most problematic one: the class imbalance problem. This occurs when there are not enough images of each species, while the number of each class should be around the same to guarantee a stable behaviour of the models. They conducted multiple experiments with distinct databases: unbalanced, balanced, images with animal in foreground, and animals manually segmented. They used CNN to classify 26 species with 6 different architectures (AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2014), GoogLenet (Szegedy et al.,

2015), Resnets: ResNet-50, ResNet-101, ResNet-152 (He et al., 2016)) and 2 additional ones with fine-tuning (AlexNet,GoogLenet). They concluded that the accuracy is better when data is balanced, and that results are better when empty images are removed or when species are segmented. The class imbalance problem could happen when there are rare species to classify (Beery et al., 2020), multiple empty images (Yang et al., 2021) or background variations (Kellenberger et al., 2018).

Thus, having enough training images is important to be able to build models able to detect and classify species correctly. The quantity of training images is also important as shown by (Shahinfar et al., 2020), if the database is balanced. According to them, 150-500 images per class is sufficient to obtain correct classification accuracy.

In our work, to handle this problem of class imbalance, we did data augmentation with horizontal flip on randomly selected images and we selected only relevant classes with enough bounding boxes detected (at least 400) and we remove empty images from our data set.

Background variation issue. When we have to work with frames from different camera traps at different places, the background variation issue happens. The difficulty is to be able to construct a model of detection and classification generalized for all localizations and new environments (modification of background or lighting conditions). In Beery et al. (2018), the authors studied the generalization of these models to be able to recognise the same species in the same region but with different camera trap backgrounds. They considered two data sets, one where training and testing images are from same location and one with different locations. For the detection step, they used

pre-trained Faster R-CNN model with two different backbones (ResNet-101 and Inception-ResNet-v2 (Szegedy et al., 2017)). They found that the model outperforms on data set with images in train and test from same location. However, when a new background appears, the results are badly affected. The background variation issue is not considered in our work, the train, validation and test data sets contain images from same locations (i.e. the different camera traps installed in the Park are present in both the learning and test sets).

Empty images. Another important problem is to deal with empty images, without any species on it. Indeed, if a movement is detected, the camera trap starts recording a video during a time-lapse fixed according to the camera settings. In many cases, the recorder video may be empty if the camera detected a movement of a branch or if the animal goes through the video quickly. Empty images biased results of CNN. To avoid having too many empty images and work only with images containing species, multiple softwares are developed to distinguish empty images (Tacka et al., 2016; Wei et al., 2020; Yousif et al., 2019) and allow to reduce time and costs instead of checking images one by one manually. As in most of previous works, we had to face to many empty videos in our data and we exploited the MegaDectector capabilities in the first stage to remove empty videos.

3. Material and methods

This section presents the 3-step process that we developed to detect, identify and count relevant species from camera trap videos collected by the Park National du Mercantour (PNM).

3.1. Material

3.1.1. Collecting data

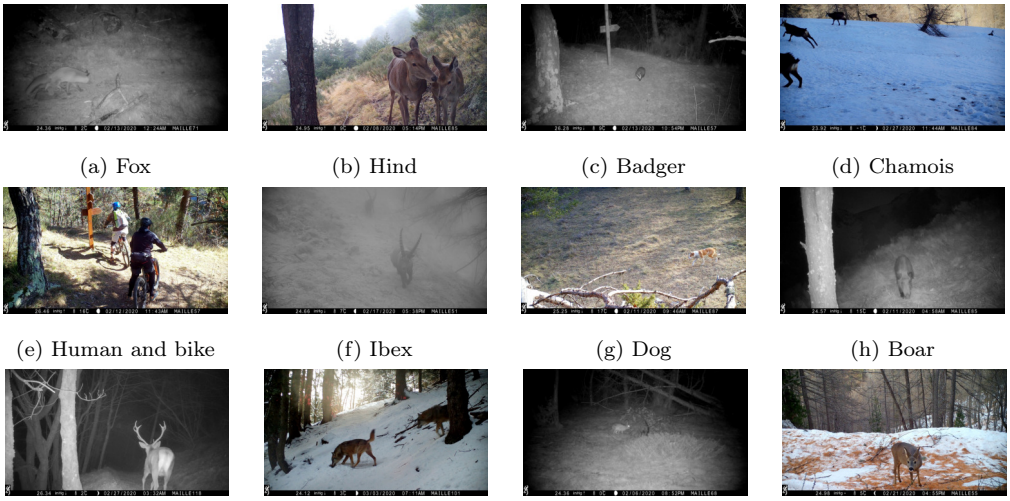
The PNM, one of the 11 national French parks, is located in *Region Sud* in France and covers an area of 1801 km^2. The highest peak of the park has an elevation of 3,143 m and is located less than 50 km from the sea. Located at the crossroads of multiple climatic, geological and altitudinal influences, the PNM is made up of a mosaic of natural environments whose extreme diversity explains the exceptional richness of fauna and flora. In order to monitor and protect the fauna, the PNM has installed 43 camera traps in "Vallée de la Roya" and "Vallée de la Vésubie" (Figure 1). When a movement is detected by a camera trap, it records a video or takes a picture. The video duration depends on the period of the day: day videos last approximately 30 seconds whereas night videos last 20 seconds. Ecologists from PNM teams referenced manually every detection from February to April 2020.

Figure 1: Map of camera traps in PNM

3.1.2. Study population

We considered in this work only camera traps which record videos. It concerns 1,744 annotated non empty videos from 35 different camera traps.

Notice that 4% of videos have multiple relevant classes on the same sample. There are 31 relevant classes present on videos: human, chamois, deer, hind, stag, fox, badger, wolf, hare, dog, boar, bike, ibex, marten, car, mountain hare, pigeon, squirrel, blackbird, jay, sparrowhawk, thrush, tengmalm's owl, wood sandpiper, owl, genette, chaffinch, weasel, lizard and butterfly. Then, each whole video is split into images all 5 tenth of a second with a resolution of 1920×1080 pixels. We obtain 87,839 images associated to one or more relevant classes (Figure 2).



(a) Fox (b) Hind (c) Badger (d) Chamois

(e) Human and bike (f) Ibex (g) Dog (h) Boar

(i) Stag (j) Wolf (k) Hare (l) Deer

Figure 2: Example of camera trap images

3.2. Methods

3.2.1. Labeling images

To train a model able to identify and count relevant classes on video, we setup a 3-step process explained in detail in step 1 of Figure 3. In a pre-processing step, MegaDetector (Beery et al., 2019) is used to locate using

bounding boxes the detected individuals. This model is classically used now by ecologists to find animals, people, and vehicles (cars, trucks and bicycles) on their images. Notice that the MegaDetector model is used here not to identify animals but just to detect them. It allows to do a first filter on their data and focus only on the animal images because human and vehicles images are not relevant for them. The accessibility of code is free and multiple organisations all over the world use it. It is trained on several hundred thousand bounding boxes from a variety of ecosystems among which public data from WCS Camera Traps, NACTI (North American Camera Trap Images), and Island Conservation Camera Traps. After applying this model on our images (during 8 hours with one GPU Nvidia Tesla V100 32Go), we obtain the coordinates of bounding boxes that we can associate with the labels of each image provided by the National Park agents. To retain the bounding boxes that we will use in the second step, the threshold on the detection confidence of the bounding boxes is fixed to 90%. It is the best confidence threshold of the bounding box detection to obtain accurate object detections. We have evaluated values between 0.5 and 0.9 by step 0.1 on the training and validation sets. We found that a value of 0.9 has the best trade-off between missing detection (with a confidence threshold higher than 0.9) and too many false detections (with a confidence threshold lower than 0.9).

When multiple labels are associated with one image (35,803 images are associated with one label while 2,355 images are associated with 2 or 3 labels), the images had to be processed manually. In this case, each image with more than one label is checked in order to know which label corresponds to which bounding box coordinates. The remaining images are classified as empty

since MegaDetector detects nothing.

Most of the relevant classes are detected by MegaDetector, among 31 relevant classes only 3 relevant classes are not detected due to their small size (weasel, lizard and butterfly). In order to have enough images for training our object detection model, we restricted ourselves to 13 relevant classes (human, chamois, deer, hind, stag, fox, badger, wolf, hare, dog, boar, bike, ibex) which are correctly detected by MegaDetector.

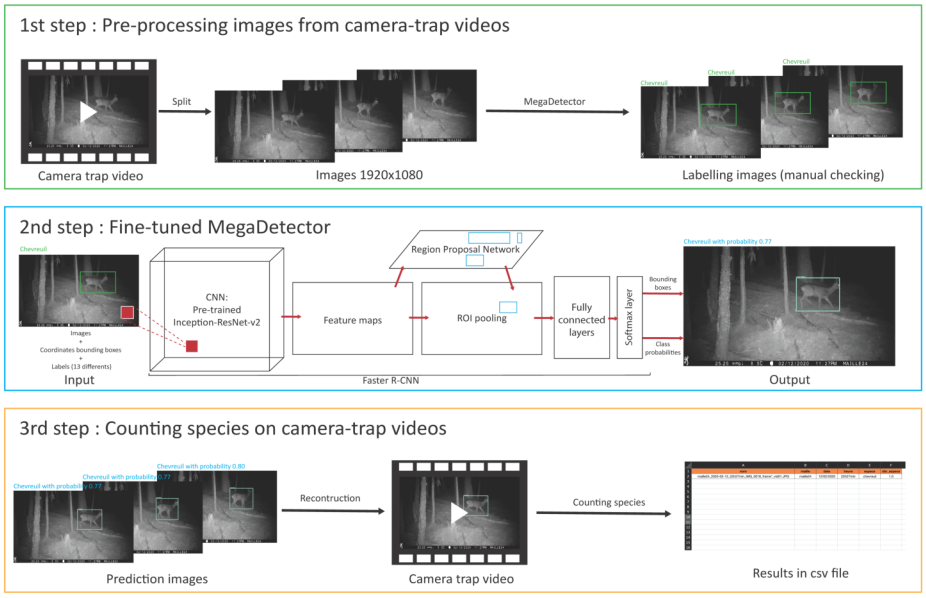


Figure 3: Process developed for analysis camera trap videos

3.2.2. Transfer learning and fine tuning

In second the step of the process (see Figure 3), we fine-tuned the object detection model to allow it to classify the detected individuals to one of the considered species. Fine tuning is a type of transfer learning (Yosinski et al., 2014). It consists in freezing a part of the current model already trained on

another data set, and retrain the last fully connected layers of the network with a new, randomly initialized final layer providing the predictions. It enables to learn new classes, which were not yet learned by the originally pre-trained model. This method helps to reach high accuracy and reduces model training time by avoiding training the entire model from scratch. (Willi et al., 2018) corroborates that transfer learning improves the model performance and outperforms training from scratch, especially when the data set available is small.

We used MegaDetector v4.1 (release 2020.04.27) as our pre-trained model. There are several advantages to use this pre-trained model: it was trained on a variety of data sets from different locations, with different species, and it shares common classes with our own model (humans in both daytime and nighttime, animals and vehicles). To begin with, we have frozen the first part of MegaDetector model, i.e. we restored the entire pre-trained feature extractor, and have only retrained the last layers (the bounding box and class prediction heads) for our own relevant classes.

3.2.3. Object detection model

An object detection model is able to locate and classify species on images from camera traps. Deep learning methods, such as CNNs (LeCun et al., 2015), are mainly used to accomplish this task, since they have shown excellent performances on image recognition. An Artificial Neural Network is composed of multiple layers, each layer is defined by a set of neurons and the connections of these neurons to the previous layer, these connections or weights are optimized through several iterations of gradient descent technique, also called backpropagation. The first "layer", called "input layer",

corresponds to the raw pixels of the image. The last layer, called "output layer", outputs the predictions of both the coordinates of bounding boxes and the associated probability for each box to belong to each class. In a Convolutional Neural Network (CNN), several of the hidden layers (i.e. neihter the input one nor the output one) are convolutional layers. A convolution layer is a specifically structured hidden layer, made of one unique neuron replicated (as many times as the layer size requests it). The weights of this single replicated neuron can then be interpreted as a convolution filter (whereby the name). Learning these convolution layers leads thus to learn convolution filters which extract different image features (e.g. edges, corners, textures, animal parts and so on). The more the number of layers, the more the model is "deep" and the more it is learning complex (visual) features. It exists 2 different types of object detection algorithms: nowadays the most popular for camera trap images are models based on region proposals such as Faster R-CNN (Ren et al., 2015), Mask R-CNN (He et al., 2017), Context R-CNN (Beery et al., 2020), or models based on regression such as YOLO (Redmon et al., 2015) and SSD (Liu et al., 2016).

In this work, we fine-tuned MegaDector composed of Faster R-CNN architecture as object detection model with backbone Inception-ResNet-v2 (Szegedy et al., 2017). Faster R-CNN is a region-based object detection algorithm. It works in two steps. The first step consists in Region Proposal Network (RPN) in order to predict where in an image a potential species could be, without knowing what kind of species it is. Then, the second step consists in applying Region-of-Interest (RoI) pooling from each RPN. RoI Pooling is used to merge multiple overlapping detections then resulting into

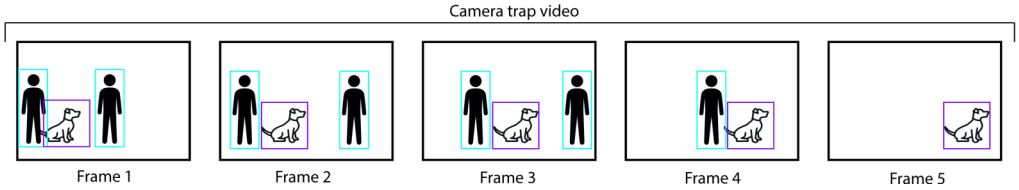
fixed-size windows of features which are then passed into two fully-connected layers to obtain the class label prediction and refine the location prediction.

3.2.4. Counting relevant classes

The final step, illustrated in the Figure 3, aims at determining how many relevant classes are present on camera trap videos based on bounding boxes detected. We fixed the confidence threshold on bounding box detections to 90%. For each camera trap images, if the detection score is under the confidence threshold, we consider the images as empty. If all frames of a video are detected as empty, then we conclude the video is empty.

Our method to estimate relevant classes is detailed in Figure 4. The method is based on bounding boxes detected by our model. Firstly, the easiest way, consists in retaining maximum number of bounding boxes detected in each frame of a video sequence, no matter which relevant classes are detected, we count only the individuals. Secondly, the more focused version of this method takes into account the classification results additionally to the detection results. It corresponds to retaining the maximum number of bounding boxes by relevant classes detected on one frame from a video sequence, in this case we count relevant classes. It is a challenging way but it allows to obtain more accurate results. As an example, in Figure 4, there are 3 individuals whose 2 are humans and 1 is a dog.

In order to evaluate our method to count relevant classes on camera trap videos, we selected 52 videos split into two sets of 24 and 28 videos respectively. Among the set of 24 videos, some frames had been used during the second step of fine-tuning MegaDetector. We thus identify these videos as the so called "train videos". To avoid introducing a bias in the counting



Basic method: there are 3 individuals (maximum bounding boxes detected on one image by individuals). Evolve method: there are 2 humans and 1 dog (maximum bounding boxes detected on one image by relevant classes).

Figure 4: Example of our method to count relevant classes on camera trap videos

process evaluation, we have added the set of 28 brand new videos (never used for training) to better assess the performance of our counting approach. They are hence called "new videos" among which 4 videos are empty. The "new videos" provided by PNM were not annotated, we manually annotated these videos (labels and count) helped by PNM team. In both cases the selection of videos was done manually, we tried to select heterogeneous videos which represent all relevant classes with variety of locations of camera traps, weather (fog, rain, snow) and moment of the day (day, night, dawn, twilight). Videos could contain one or multiple relevant classes (same or different) with different conditions (far away or hidden). Thanks to this variety of videos, we can challenge our own model.

4. Experiment and results

This section presents the main results of the application of our approach on the data provided by the Parc National du Mercantour.

4.1. Experiment

4.1.1. Data sets

We have 37,424 single images for 52,470 bounding boxes from day and night conditions and this for our 13 relevant classes. To evaluate the robustness of our model, we split these images into 3 data sets: training, validation and test. The train data set corresponds to 80% of data (29,976 frames), the validation data set of 10% of data (3,705 single images) and the test data set of 10% of data (3,743 single images). Notice that, since it is possible to have multiple relevant classes on a same image, the number of single images is different from the number of images by relevant classes (Table 2) and from the number of bounding boxes detected by relevant classes (Table 1). Furthermore, images are not entirely randomly affected in each data set since we considered that an image could be only in one data set (i.e. training, or validation, or test). For instance, if they are more than one relevant class on an image, the image (with its labels and the bounding boxes associated) is assigned to only one of these 3 data sets. The aim is to avoid identical image repetitions in multiple data sets in order to not bias our results. In addition, to increase the ability of our model to generalize with more robust features learned (reducing overfitting) and to generate additional training data, we did data augmentation. We generated new training samples by applying an horizontal flip to randomly selected original images. Nevertheless, we faced the problem of unbalanced data sets: the human is over-represented in comparison to other relevant classes.

Table 1: Number of bounding boxes by relevant classes in the data set

Species	**Train**	**Validation**	**Test**	**Total**
Human	14775	1726	1826	18327
Chamois	6270	774	779	7823
Deer	4893	655	594	6142
Hind	4811	578	554	5943
Stag	3501	481	454	4436
Fox	3439	434	415	4288
Badger	983	115	118	1216
Wolf	767	84	102	953
Dog	586	65	64	715
Boar	565	81	66	712
Hare	563	81	88	732
Bike	553	66	90	709
Ibex	378	44	52	474
Total	42084	5184	5202	**52470**

Table 2: Number of images by relevant classes in the data set

Species	**Train**	**Validation**	**Test**	**Total**
Human	9086	1080	1160	11326
Chamois	3816	466	472	4754
Deer	4187	549	515	5251
Hind	3429	419	395	4243
Stag	2706	360	360	3426
Fox	3399	426	409	4234
Badger	983	114	118	1215
Wolf	722	78	97	897
Dog	585	64	64	713
Boar	478	64	58	600
Hare	555	80	86	721
Bike	535	65	84	684
Ibex	351	41	49	441
Total	30832	3806	3867	**38505**

4.1.2. Experimental setup and architecture

We used TensorFlow 1 Object Detection API (version 1.12.0) and python language to fine-tune the MegaDetector model on a machine with one GPU Nvidia Tesla V100 32 Go. We fine-tuned MegaDetector model in order to obtain accurate results from our data rather than starting from scratch. The fine-tuning is applied until detection step, only the 4 last layers (detection and classification) are trained to our own 13 relevant classes. It lasted for 16 hours and 12,000 epochs with an evaluation at each 500 epochs. The architecture of MegaDetector is adjusted to our study but most of its settings remain unaltered. In our work, just as MegaDetector architecture, the feature extractor is Faster R-CNN with Inception-ResNet-v2, the output layer is a Softmax activation function and the evaluation protocol is "COCO detection metrics".

Compared to MegaDetector, few parameters modifications are tested: number of batches, optimizers, learning rate values and image size. We identified the optimal parameters using the validation data set. We established that the optimal number of batches is 14 and the optimal size of the images is 480x270 (i.e. original size divided by 4). The best algorithm optimisation is Adam with initial learning rate value of 1e-5. Notice that our algorithm can detect 100 bounding boxes on each image, which could be useful in the case of multiple relevant classes present at that moment.

Finally, monitoring training and validation losses, we were able to do an early stopping and we selected the model saved at 10,000 epochs (Figure 5).

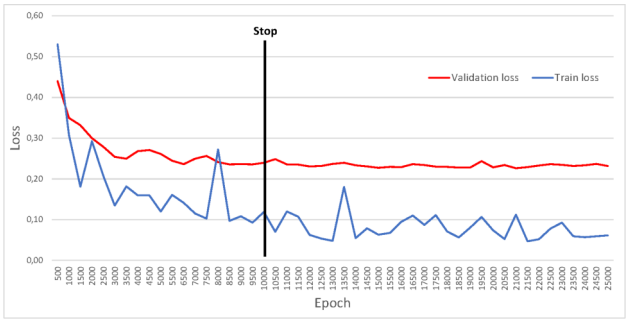


Figure 5: Loss functions

4.2. Results

4.2.1. Detection and classification

Table 3 presents the results obtained with our model on validation and test data sets. To evaluate performance of our object detection model, for both detection and classification steps, we used mean Average Precision (mAP) metric. This metric is based on Intersection-over-Union (IoU) measure. It gives the overlap between the ground truth bounding box and the predicted bounding box. It is commonly admitted that a IoU value at least equal to 0.50 validate a detection. The mAP corresponds to the average of Average Precision (AP) values over all relevant classes for IoU from 0.50 to 0.95 with a step size of 0.05. Furthermore, we also considered mAP at IoU=0.50 and at IoU=0.75.

It is first interesting to notice that the model results on test data set are very close to validation data set, validating in turn the robustness of our model. Secondly, the evaluation model achieves around 74% mAP, 97% mAP at IoU=0.50 and 89% mAP at IoU=0.75. Human is the easiest species to detect and classify with around 81% mAP, we expected this result because

we have lot of images with human individuals under different conditions. It is different for dog and hare, which are the hardest to detect and classify with around 66% mAP. These are satisfactory results accounting for the difficulty of the task and the amount of images for these specific classes. Results of mAP per class can be seen in Figure 6.

Table 3: mAP across validation and test data sets

Metrics	**Validation**	**Test**
mAP	74,11%	73,92%
mAP at IoU=0.50	96,79%	96,88%
mAP at IoU=0.75	89,32%	89,24%

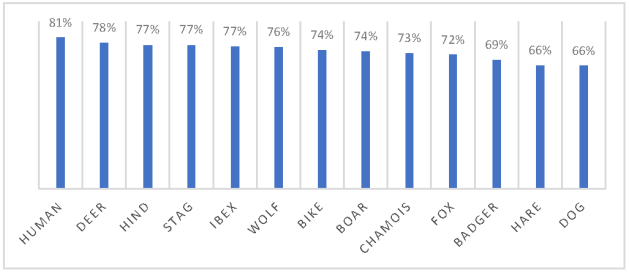
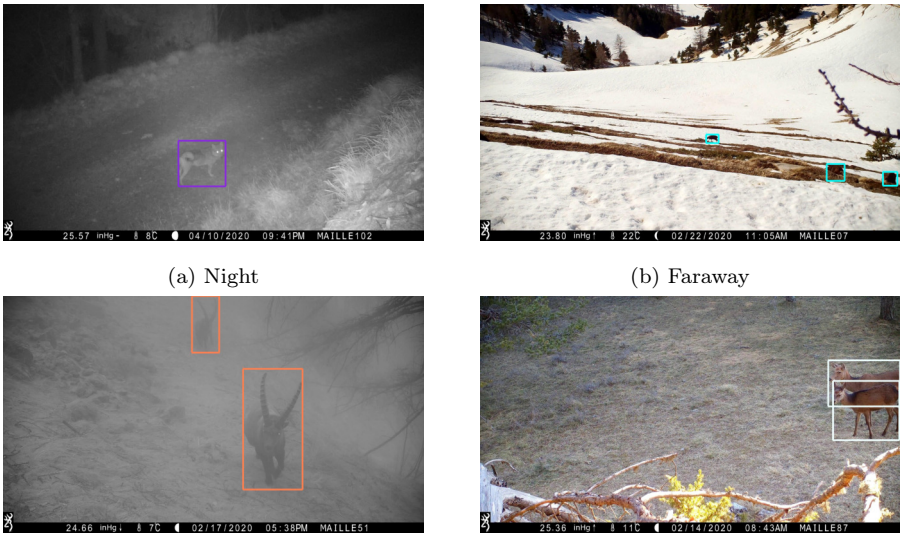


Figure 6: mAP by relevant classes across test data set

Figure 7 shows model capacity to detect relevant classes in multiple situation: far away from lens, with fog, by night and with occlusion. We observed that overall, our model achieved best performance in detection step. As regards to classification step, our model also presents satisfactory results especially for big species and species over-represented like human. Figure 8 shows that good detection and classification of relevant classes depend mostly

on the context of the image. For example, we suppose hare videos by night is more frequent than by day, so it is more difficult to detect them by day. Moreover, it is less frequent to see dogs' back than to see them by the side on camera trap videos. We also notice, that distinguishing deer from hind or stag is not easier for deep learning models than it is for humans. In the case of wolf, we observe that detecting and classifying a wolf far away from the lens represents no difficulty for our model, even with the head of a chamois in its mouth.



(a) Night

(b) Faraway

(c) Fog

(d) Occlusion

Figure 7: Relevant classes detection in different conditions (detection threshold on the bounding boxes confidence of 90%)

4.2.2. Count of detected individuals

Figure 9 shows the results we obtained to predict the number of individual presents on 52 camera trap videos. These results are only based on detection results. Overall, for 62% of videos we predicted the exact number

(a) Correct detection and classification

(b) Correct detection but misclassified

Figure 8: Examples of results of detection and classification (detection threshold on the bounding boxes confidence of 90%)

of individuals and for 87% of them the count predicted is either exact or ± 1 unit. Detection algorithm works well on both data set of videos, we obtained similar results for "train videos" and "new videos". As regards to empty videos, for 50% of them we predicted exactly to be empty videos. The 50% remaining are predicted with difference of one unit.

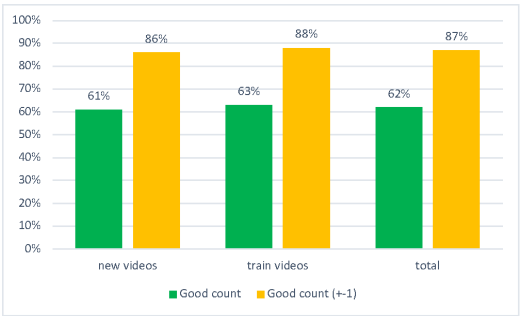


Figure 9: Results of counting individuals detected on camera trap videos

4.2.3. Count of detected relevant classes

The results obtained to predict the number of relevant classes present on camera trap videos are showed in Figure 10. Due to the difficulty of the task, to count relevant classes based on detection and classification results, lower results are obtained in this experiments than in the previous experiments. We obtained 81 predictions of relevant classes associated to count for really 58 relevant classes associated to count. Generally, for both "train videos" and "new videos", we correctly predicted 38% relevant classes with exactly count and for 48% of videos we correctly predicted relevant classes with either exact count or ± 1 unit. Here, results obtained for "train videos" are better than "new videos". For 55% of "train videos", we predicted the exact number of relevant classes whereas for "new videos" is 28%. Finally, concerning

empty videos, we reached same results as in the previous experiments, for 50% of them we predicted them correctly to be empty videos. The 50% remaining are predicted with difference of one unit, which remains a totally acceptable count given the wildlife monitoring objectives of the Parc National du Mercnatour.

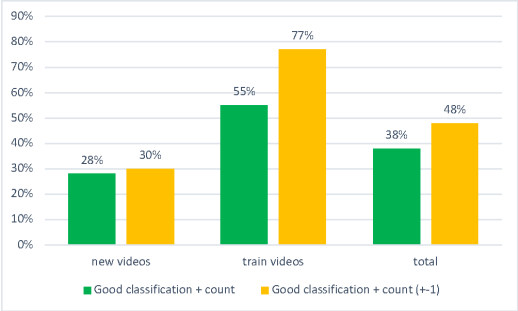


Figure 10: Results of counting relevant classes detected on camera trap videos

5. Discussion

The environment preservation field becomes more and more interested in benefiting from recent advances in AI. Deep learning methods in particular to avoid to ecologists to do tedious tasks (Tuia et al., 2022). In this paper, we demonstrated that it is possible to locate, identify and estimate population of relevant wildlife species from camera trap videos thanks to deep learning models. Firstly, we split videos into images and label them. Then we fine-tuned an object detection model for our problem. We obtained convincing results for detection and classification steps. On the one hand, our work demonstrates how crucial is an accurate selection for the hyperparameters of the deep model. These parameters (batch size, learning rate, optimiser,

method of data augmentation and size of images) have been selected on the training data by splitting them into learning and validations sets. The results on test data have shown the robustness of the hyperparameter selection. On the other hand, transfer learning, i.e. exploiting databases composed of variety of species from different locations to pre-train our model, improved significantly the results. In last step of our work, we considered the question of counting the species individuals and our counting method, based on the maximum number of bounding boxes detected on one frame, obtained satisfactory results for the National Park objectives. Our model, DeepWILD, detects and classify 13 relevant classes with different conditions (night, fog, snow, rain, far away or close to lens, occlusion) from different locations of camera traps. It is a first step to follow species in the Parc National du Mercantour. The model can distinguish empty videos from wildlife or human or vehicles. The performances for detection and classification steps of our model are bounded to the quantity of images available during training phase and the acquisition conditions of images. The method used to count relevant classes allows to have a first good estimation of the population present on camera trap videos. These tasks still remain a challenge: using object detection models for camera trap videos with variety of images and especially counting species. Few of the articles reviewed address these issues. Nevertheless, this work brings new ways to estimate wildlife population and it represents a great support for ecologists, that will save a huge amount of their time to analyse plenty of videos each month.

Limitations. At the moment, the originality of DeepWILD is that our model covers all the critical tasks: detection, classification, and counting species on

videos from camera traps. However, to train such deep models, the amount of training data required has to be of several thousands (not to say more). Even though most of camera traps available in PNM (81%) capture videos rather than images, we have decided to work at the frame level to get enough training/validation/test data, splitting thus the videos with short time-lapse into images. In order to reduce bias in these data sets, when there are multiple species present in a frame, we avoid having the same image present in both train, validation and test sets. An image can only be in one of these sets. The impact of this data set curation is negligible on the final size of the training data. However, if this process solved the question of the size of the training set, it introduced new problems such as the generalisation of our model. Indeed, since the time-lapse is short, consecutive images from the same video show very high visual similarities. If consecutive frames are then distributed in the training, validation, and test sets, the impact on the generalisation power of our model is immediate.

Although, the quantity of videos provided by the PNM allowed us to build a large enough data set and to design a model able to detect 13 relevant classes by night and day, this amount of data is still not enough to train a model from scratch robust to all possible conditions. We would need more videos with different conditions (day, night, rain, fog, snow ...) and a better balance between classes to improve the results.

Finally, it is still a hard challenge to count species on camera trap videos. Our current main problem is to deal with several individuals from the same species passing in front of a camera trap with few seconds between each individual as illustrated in Figure 11. In this situation our algorithm will

consider that there is only one individual on the video while they are two but from the same species. Distinguishing that those two individuals from the same species are not the same individual passing twice in front of the camera trap, is also a challenge for human so we need to find how to integrate this higher level temporal consistency in the model in order to solve this frequent configuration.

DeepWILD is not yet fully deployed by PNM because the main objective was to specifically monitor the wolf presence in PNM, but automatic detection, classification, then counting of wolves are not accurate enough at the moment. We would need more images of wolves to improve the results. However, in its actual stage, it is already used to filter data, to remove empty videos and videos with humans in order to avoid GDPR issues, and to easily and quickly focus on animal videos. The final deployment is still expected in a soon future.

Future work. As for further directions, we ambition to improve our detection and classification performances by either increasing the number of annotated videos thanks to labelling image application, considering images rather than only videos from camera traps or testing other deep learning models such as Mask R-CNN or Context R-CNN. The combination of these 3 approaches could also be considered to further improve the detection and classification performance in a ensemble decision flavor. Furthermore, regarding new species passing in front of the camera traps (for example jackal or lynx) or even young animals (for example young wild boar or wolf cub), which is a challenging situation, we could use models as Few-shot learning, One-shot learning or Zero-shot learning. We could also consider to add an extra class

named "other", gathering all additional relevant classes provided by PNM (15 other relevant classes than the 13 we have worked with) and that we did not study in the current work. Finally, a future work concerning the counting of species could be to use tracking methods, such as Multi-Object Tracking (MOT) into offline tracking, to be able to follow species on camera traps videos and therefore improve our counting method. We could also consider more precise segmentation methods such as, for instance, the one proposed in (Giraldo-Zuluaga et al., 2019).

(a) Frame 2 (b) Frame 3 to 40 (c) Frame 41

Figure 11: Example of issue meet to count relevant classes on camera trap videos

Acknowledgements

This work was made possible thanks to the collaboration with Parc National du Mercantour, which provided both the ecological question and the camera trap data. The authors want to thank in particular Nathalie Siefert and Stéphane Combeaud. This work was supported by the French government, through the UCAJEDI and the 3IA Côte d'Azur Investment in the Future projects managed by the National Research Agency (ANR) under the reference numbers: ANR-15-IDEX-01 and ANR-19-P3IA-0002. The authors are also grateful to the OPAL infrastructure from Université Côte d'Azur and the Université Côte d'Azur's Center for High-Performance Computing

for providing resources and support.

References

D. Tuia, B. Kellenberger, S. Beery, B. R. Costelloe, S. Zuffi, B. Risse, A. Mathis, M. W. Mathis, F. van Langevelde, T. Burghardt, et al., Perspectives in machine learning for wildlife conservation, Nature communications 13 (2022) 1–15.

J. Wäldchen, P. Mäder, Machine learning for image based species identification, Methods in Ecology and Evolution 9 (2018) 2216–2225. doi:https://doi.org/10.1111/2041-210X.13075.

R. Chen, R. Little, L. Mihaylova, R. Delahay, R. Cox, Wildlife surveillance using deep learning methods, Ecology and Evolution 9 (2019) 9453–9466. doi:https://doi.org/10.1002/ece3.5410.

M. Vargas-Felipe, L. Pellegrin, A. A. Guevara-Carrizales, A. P. López-Monroy, H. J. Escalante, J. A. Gonzalez-Fraga, Desert bighorn sheep (ovis canadensis) recognition from camera traps based on learned features, Ecological Informatics 64 (2021) 101328.

S. Schneider, G. W. Taylor, S. C. Kremer, Deep learning object detection methods for ecological camera trap data, in: 2018 15th Conference on Computer and Robot Vision (CRV), 2018. doi:https://doi.org/10.1109/CRV.2018.00052.

A. C. Ferreira, L. R. Silva, F. Renna, H. B. Brandl, J. P. Renoult, D. R. Farine, R. Covas, C. Doutrelant, Deep learning-based methods for indi-

vidual recognition in small birds, Methods in Ecology and Evolution 11 (2020) 1072–1085. doi:https://doi.org/10.1111/2041-210X.13436.

K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

M. S. Norouzzadeh, D. Morris, S. Beery, N. Joshi, N. Jojic, J. Clune, A deep active learning system for species identification and counting in camera trap images, Methods in Ecology and Evolution 12 (2020) 150–161. doi:https://doi.org/10.1111/2041-210X.13504.

S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, volume 28, 2015. URL: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

S. Beery, A. Agarwal, E. Cole, V. Birodkar, The iwildcam 2021 competition dataset, 2021. doi:https://doi.org/10.48550/arXiv.2105.03494.

F. Sarwar, A. Griffin, P. Periasamy, K. Portas, J. Law, Detecting and counting sheep with a convolutional neural network, in: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018. doi:https://doi.org/10.1109/AVSS.2018.8639306.

R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014. doi:https://doi.org/10.1109/CVPR.2014.81.

B. Xu, W. Wanga, G. Falzon, P. Kwan, L. Guoa, G. Chen, A. Tait, D. Schneider, Automated cattle counting using mask r-cnn in quadcopter vision system, Computers and Electronics in Agriculture 171 (2020) 105–300. doi:https://doi.org/10.1016/j.compag.2020.105300.

K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi:https://doi.org/10.1109/CVPR.2016.90.

D. Levy, Y. Belfer, E. Osherov, E. Bigal, A. P.Scheinin, H. Nativ, D. Tchernov, T. Treibitz, Automated analysis of marine video with limited data, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018. doi:https://doi.org/10.1109/CVPRW.2018.00187.

N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: 2017 IEEE International Conference on Image Processing (ICIP), 2017. doi:https://doi.org/10.1109/ICIP.2017.8296962.

T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

L. Zhang, H. Gray, X. Ye, L. Collins, N. Allinson, Automatic individual pig detection and tracking in surveillance videos, 2018. doi:https://doi.org/10.48550/arXiv.1812.04901.

J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Advances in Neural Information Processing Systems, volume 29, 2016. URL: https://proceedings.neurips.cc/paper/2016/file/577ef1154f3240ad5b9b413aa7346a1e-Paper.pdf.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C.Berg, Ssd: Single shot multibox detector, in: Computer Vision – ECCV 2016, Springer, Cham, 2016, pp. 21–37. doi:https://doi.org/10.1007/978-3-319-46448-0_2.

K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014. doi:https://doi.org/10.48550/arXiv.1409.1556.

M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packera, J. Clune, Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning, Proceedings of the National Academy of Sciences 115 (2018) E5716–E5725. doi:https://doi.org/10.1073/pnas.1719367115.

A. G. Villa, A. Salazar, F. Vargas, Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks, Ecological Informatics 41 (2017) 24–32. doi:https://doi.org/10.1016/j.ecoinf.2017.07.004.

A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, volume 25, 2012.

URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. doi:https://doi.org/10.1109/CVPR.2015.7298594.

S. Beery, Y. Liu, D. Morris, J. Piavis, A. Kapoor, M. Meister, N. Joshi, P. Perona, Synthetic examples improve generalization for rare classes, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2020.

D.-Q. Yang, T. Li, M.-T. Liu, X.-W. Li, B.-H. Chen, A systematic study of the class imbalance problem: Automatically identifying empty camera trap images using convolutional neural networks, Ecological Informatics 64 (2021). doi:https://doi.org/10.1016/j.ecoinf.2021.101350.

B. Kellenberger, D. Marcos, D. Tuia, Detecting mammals in uav images: Best practices to address a substantially imbalanced dataset with deep learning, Remote Sensing of Environment 216 (2018) 139–153. doi:https://doi.org/10.1016/j.rse.2018.06.028.

S. Shahinfar, P. Meek, G. Falzona, "how many images do i need?" understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring, Ecological Informatics 57 (2020). doi:https://doi.org/10.1016/j.ecoinf.2020.101085.

S. Beery, G. van Horn, P. Perona, Recognition in terra incognita, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.

C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 31, 2017. doi:https://doi.org/10.1609/aaai.v31i1.11231.

J. L. Tacka, B. S.West, C. P.McGowan, S. S.Ditchkoff, S. J.Reeves, A. C.Keever, J. B.Grand, Animalfinder a semi-automated system for animal detection in time-lapse camera trap images, Ecological Informatics 36 (2016) 145–151. doi:https://doi.org/10.1016/j.ecoinf.2016.11.003.

W. Wei, G. Luo, J. Ran, J. Li, Zilong: A tool to identify empty images in camera-trap data, Ecological Informatics 55 (2020). doi:https://doi.org/10.1016/j.ecoinf.2019.101021.

H. Yousif, J. Yuan, R. Kays, Z. He, Animal scanner: Software for classifying humans, animals, and empty frames in camera trap images, Ecology and Evolution 9 (2019) 1578–1589. doi:https://doi.org/10.1002/ece3.4747.

S. Beery, D. Morris, S. Yang, Efficient pipeline for camera trap image review, 2019. doi:https://doi.org/10.48550/arXiv.1907.06772.

J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: Advances in Neural Information Processing systems, volume 27, 2014. URL: https://proceedings.neurips.cc/paper/2014/file/375c71349b295fbe2dcdca9206f20a06-Paper.pdf.

M. Willi, R. T. Pitman, A. W. Cardoso, C. Locke, A. Swanson, A. Boyer, M. Veldthuis, L. Fortson, Identifying animal species in camera trap images using deep learning and citizen science, Methods in Ecology and Evolution 10 (2018) 80–91. doi:https://doi.org/10.1111/2041-210X.13099.

Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444. doi:https://doi.org/10.1038/nature14539.

S. Beery, G. Wu, V. Rathod, R. Votel, J. Huang, Context r-cnn: Long term temporal context for per-camera object detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. doi:https://doi.org/10.1109/CVPR42600.2020.01309.

J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

J.-H. Giraldo-Zuluaga, A. Salazar, A. Gomez, A. Diaz-Pulido, Camera-trap images segmentation using multi-layer robust principal component analysis, The Visual Computer 35 (2019) 335–347.