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
Charles Bouveyron
Frédéric Precioso

October 4, 2022

Abstract

Videos and images from camera traps are more and more used by ecologists to estimate the population of species on a territory. Most of the time, it is a laborious work since the experts analyse manually all this data. It takes also a lot of time to filter these videos when there are plenty of empty videos or with humans presence. Fortunately, deep learning algorithms for object detection could 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. We developed a 3-parts process to analyse camera trap videos. At the first stage, after splitting videos into images, we annotate images by associating bounding boxes to each label thanks to MegaDetector algorithm. Then, we extend MegaDetector based on Faster R-CNN architecture with backbone Inception-ResNet-v2 in order to not only detect the 13 species considered but also to classify them. Finally, we define a method to count species based on maximum number of bounding boxes detected, it included only detection results and an evolve version of this method included both, detection and classification results. The results obtained during the evaluation of our model on the test dataset 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. Big species highly represented, like human, have highest values of mAP around 81% whereas species less represented in the train dataset, such as dog, have lowest values of mAP around 66%. As regards to our method of counting, we predicted a count either exact or ± 1 unit for 87% with detection results and 48% with detection and classification results of our video sample. 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.

Camera trap CNN Deep learning Image classification Object detection

1 Introduction

The potential of Machine Learning (ML) on wildlife conservation is more and more investigated in the last years, to address very diverse tasks such as recognizing species from different modalities (audio for birds or cetaceans, visual for animals or humans), such as tracking and pose estimation, etc. This potential has been very 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 as researchers in ecology as well as private individuals who want to detect and track animals on their property. For species preservation, ecologists need to identify species presence on a territory, estimate their quantity and their 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 using camera traps the different wildlife species present on its territory to better understand how animals live, in order also better protect them. 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 this national park. However, most of ecology researchers who currently use camera traps to monitor wildlife species in the national French parks, have to do it manually: they watch each camera trap videos and manually count the different species presence on 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 camera trap videos. 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 by doing tedious tasks.

In this study, we developed a process to filter empty images extracted from camera trap videos and labeling them thanks to preexisting object detection model and manual checking. Then, we extended a pre-trained model in order to detect our own species. Finally, we elaborated a method to count species on each camera trap videos based on bounding boxes detection.

2 Related work

In recent years, deep learning methods are increasingly used for videos and frames analyses from camera traps in order to identify and classify species (Wäldchen & Mäder, 2018). It helps ecologists to avoid doing this task manually. Convolutional neural network (CNN) is the deep learning method mainly used to do image classification, in other words to identify one or multiple species on images (Chen, Little, Mihaylova, Delahay, & Cox, 2019). Vargas-Felipe et al. (2021) propose to use not only data from their camera traps but also to augment the training sets with pictures from the web of related specices 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, CNN can localise species on different frames from a video, it is called object detection (Schneider, Taylor, & Kremer, 2018). For example, as a first step of their classification, (Ferreira et al., 2020) used Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017), a model for object detection, which automatically localises one of the three studied bird species and crops them in the images.

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 review hereafter that. For example, Norouzzadeh et al. (2020), simply considered only one class "animal" then they counted on images by summing the number of bounding boxes detected with a detection threshold on the BB confidence of 90%. In order to obtain these bounding boxes, they used a pre-trained object detection model, based on Faster-RCNN object detection algorithm (Ren, He, Girshick, & Sun, 2015) and trained their model on different camera trap datasets. 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, Agarwal, Cole, & Birodkar, 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 detection threshold on the BB confidence of 80%. 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 Vehicles videos. Sarwar, Griffin, Periasamy, Portas, and Law (2018) used it in order to detect and count sheep in a paddock to help farmers by comparing two methods, one with R-CNN

(Girshick, Donahue, Darrell, & Malik, 2014) and another one with hand crafted technique. A similar method is applied by (Xu et al., 2020) on images captured by a quadcopter but with another object detection algorithm. In this work, the authors used pre-trained model Mask R-CNN with a ResNet-101 (He, Zhang, Ren, & Sun, 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, Bewley, & Paulus, 2017) combined with RetinaNet (Lin, Goyal, Girshick, He, & Dollár, 2017) for the detection part in order to detect, classify and count the marine organisms on two different datasets, an aerial and underwater frame from marine videos. Another tracking method is used in (Zhang, Gray, Ye, Collins, & Allinson, 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, R-FCN (Dai, Li, He, & Sun, 2016) and SSD (Liu et al., 2016)) with backbone VGG16 (Simonyan & Zisserman, 2014) 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), also calculate 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. It is only used to count one unique species by frame, not multiple species.

There are multiple challenges associated to study frames of videos from camera traps: poor illumination, occlusion, complex animal pose, blurred, over-exposed, size of species, day or nighttime, animals far away from camera or too close to camera, background variation, multiple species on same images, empty images or lack of images. (Villa, Salazar, & Vargas, 2017) enumerate these different issues for species recognition in camera trap images analyses and decided to focus on the most problematic one: the class imbalance problem. This occur when there are not enough images of each species, the number of each class must be the same. They realised 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, Sutskever, & Hinton, 2012), VGGNet (Simonyan & Zisserman, 2014), GoogLenet (Szegedy et al., 2015), Resnets: ResNet-50, ResNet-101, ResNet-152 (He et al., 2016)) and 2 with fine tuning (AlexNet,GoogLenet). They conclude that performance, measure by accuracy metric, is better when data is balanced whereas unbalanced and 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, Liu, et al., 2020), multiple empty images (Yang, Li, Liu, Li, & Chen, 2021) or background variations (Kellenberger, Marcos, & Tuia, 2018).

Thus, having enough images is important to be able to build models to detect and classify species correctly. The quantity of training images is also important as is shown by (Shahinfar, Meek, & Falzona, 2020), if the database is balanced, the model accuracy is improved when there is a high number of images in the train part. According to them, 150-500 images per class is sufficient to obtain correct classification accuracy.

Regarding to the background variation issue, it happens when we have to work with frames from different camera traps at different places. The difficulty is to be able to construct a model of detection and classification generalised for all localisations and new environments (modification of background or lighting conditions). Beery, van Horn, and Perona (2018) study the generalisation of these models to be able to recognise the same species in the same region but with different camera trap backgrounds. They considered two datasets, one where training and testing images are from same location and one with different locations. For the detection part, they used pre-trained Faster R-CNN model with two different backbones (ResNet-101 and Inception-ResNet-v2 (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017)). They find that the model outperforms on dataset with images in train and test from same location. When a new background appears, the results are badly affected.

Moreover, another biggest problem is to work on empty images, without any species on it. Indeed, if a species is detected, the camera trap starts to record video during a time-lapse fixed according to the camera trap setup. Most of the time, the animal goes through the video quickly, it does not stay a long time in front of the camera. Thus, images could contain species or could be empty, there are also false triggers due to moving vegetation. Empty images biased results of CNN. To avoid having too many empty images and work only with images containing species, multiple software are developed to distinguish empty images (Tacka et al., 2016; Wei, Luo, Ran, & Li, 2020; Yousif, Yuan, Kays, & He, 2019) and allow to reduce time and costs instead of reviewing images one by one manually.

3 Material and methods

Thanks to camera trap videos from various species collected by PNM, we developed a process in 3 steps to process them and define an object detection model able to detect, identify and count species.

3.1 Material

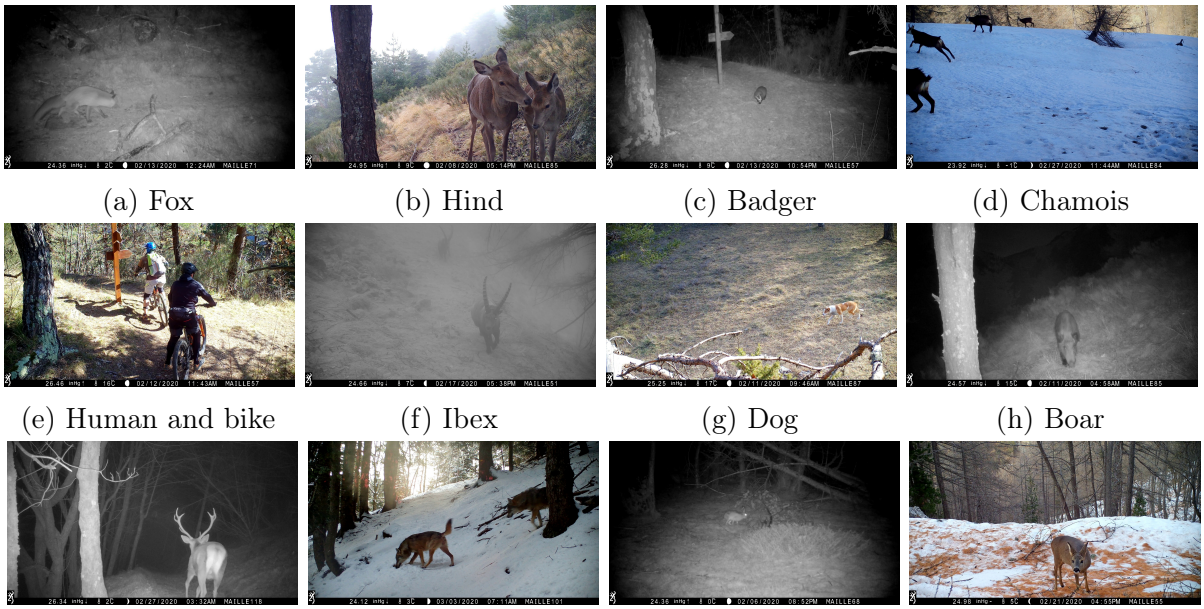
3.1.1 Collecting data

The PNM, one of 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 3143 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 starts to record a video or take a picture. The video duration depends of the day and night: commonly day videos last approximately 30 seconds whereas night videos last 20 seconds. Ecologist from PNM referenced manually every detection from February to April 2020.

Figure 1: Map of camera traps in PNM

3.1.2 Study population

We considered only camera traps which record videos, its concerned 1,744 annotated non empty videos from 35 different camera traps. 4% of videos have multiple species on same video. There are 31 species 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, the whole videos are split into images all 5 tenth of a second with resolution of 1920×1080 pixels. We obtained a sample of images of 87,839 images associated to one or more species (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 species on video, each camera trap image has to be associated to one label and to its coordinates of bounding boxes. This process is explained in detail in step 1 of Figure 3. In order to obtain the bounding boxes's coordinates, we use a model called MegaDetector (Beery, Morris, & Yang, 2019). This model is originally only able to detect people, animals and vehicles (cars, trucks and bicycles) on camera trap images. It does not identify animals, it just detects them. It was trained on bounding boxes from a variety of ecosystems. After applying this model on our images (during 8 hours with one GPU Nvidia Tesla V100 32 Go), we obtained the coordinates of bounding boxes that we can associated with the labels of each image. To retain the bounding boxes that we will use in the second step, detection threshold on the BB confidence is fixed at 90%. It is the best detection threshold on the BB confidence to obtain accurate detection.

When multiple labels are associated with one image, it is even more complicated: 35,803 images are associated with one label whereas 2,355 images are associated with 2 or 3 labels. These images are processed manually, all images with more than one label are checked in order to know which label corresponds to bounding box coordinates and images. The remaining images are classified as empty since MegaDetector detects nothing.

Most of the considered species are detected by MegaDetector, among 31 current species only 3 species are not detected due to their small size (weasel, lizard and butterfly). In order to have enough images for train our object detection model, we considered 13 species

(human, chamois, deer, hind, stag, fox, badger, wolf, hare, dog, boar, bike, ibex) which are mostly detected by MegaDetector.

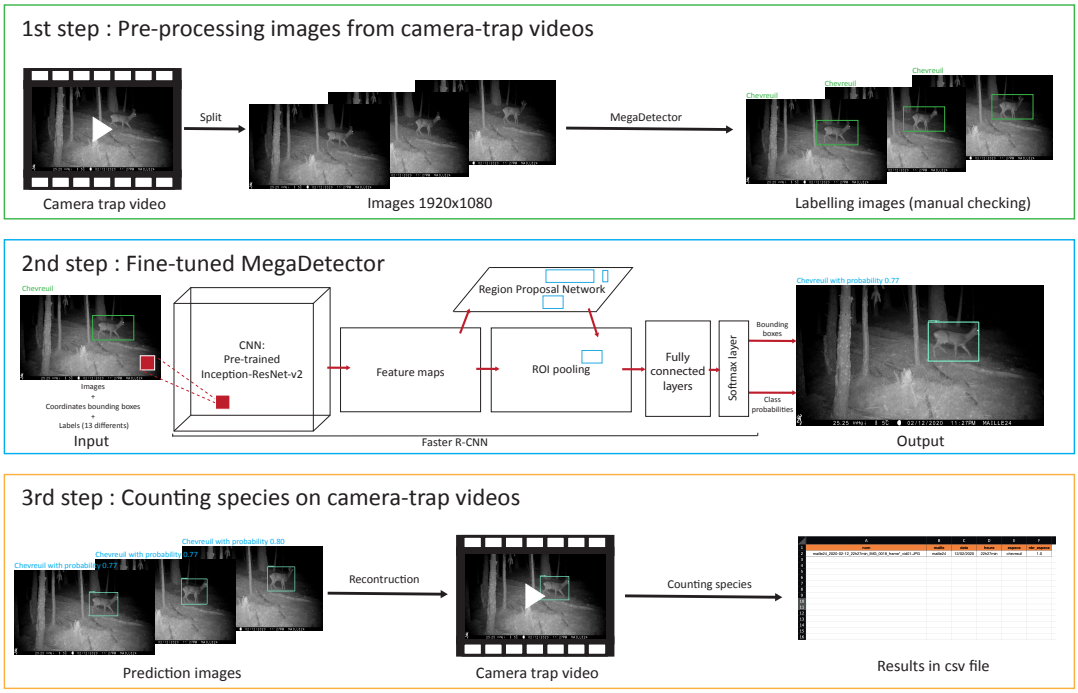


Figure 3: Process developed for analysis camera trap videos

3.2.2 Object detection model

An object detection model is able to locate and classify species on images from camera traps. Deep learning methods (LeCun, Bengio, & Hinton, 2015), such as CNN, are mainly used to accomplish this task, since they have shown excellent performances on image recognition. CNN is composed of multiple convolutional layers followed by at least one fully connected layer. Each layer is connected to the next and previous layers by weights (also called filters), these weights are corrected by backpropagation method. The first layer, called "input layer", corresponds to the raw pixels of images. The last layer, called "output layer", predicts the coordinates of bounding boxes associated to the probability for each box to belong to each class. Among the hidden layers, the layers between the input layer and output layer, convolution is applied with filters which extract different features of images (edges, corners, textures, animal parts and so on). The more the number of layers, the more the model is getting deeper and the more it is learning 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, Wu, Rathod, Votel, & Huang, 2020), or models based on regression such as YOLO (Redmon, Divvala, Girshick, & Farhadi, 2015) and SSD (Liu et al., 2016).

In this work, we used 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 predicting multiple Region Proposal Network (RPN) in order to determine where in an image a potential species could be, without knowing what kind of species is. Then, the second step consists in applying Region of Interest (ROI) pooling from each RPN. ROI Pooling is used to extract 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.

3.2.3 Transfer learning and fine tuning

In second step of Figure 3 we fine-tuned the object detection model. Fine tuning is a type of transfer learning (Yosinski, Clune, Bengio, & Lipson, 2014). It consists in freezing a part of the current model already trained and retrain the fully connected head of network with a new, randomly initialized head. It enables to learn new classes, which were not yet learned by the 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 dataset available is smaller.

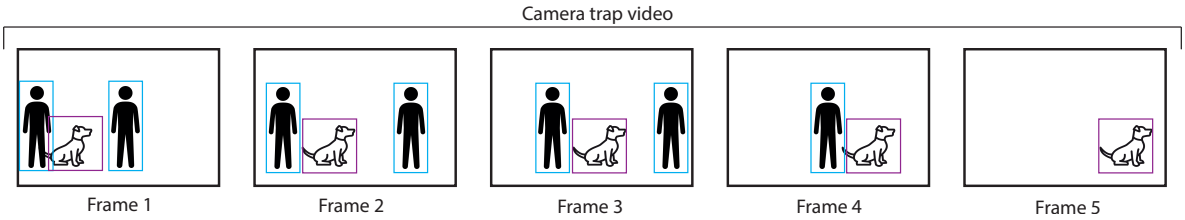
We used MegaDetector v4.1 (release 2020.04.27) as our pre-trained model. There are many advantages to use this pre-trained model: it was trained on a variety of datasets from different locations, with different species, and it shares common classes with our own model (humans in both daytime and nighttime,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

3.2.4 Counting species

We propose here in the third step of Figure 3, a method to determine how many species are present on camera trap videos based on bounding boxes detected. We fixed detection threshold on the BB confidence value of correct detection to 90%. For each camera trap images, if the detection score is under the detection threshold on the BB confidence fixed, 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 species are detailed in Figure 4. The method is based on bounding boxes detected by our model describe previously. Firstly, the easiest way, consists in retaining maximum number of bounding boxes detected in each frame of a video sequence, no matter which species are detected, we count only the individuals. Secondly, the evolution of this method takes into account the classification results additional to

detection results from our previous model. It corresponds to retaining maximum number of bounding boxes by species detected on one frame from video sequence, in this case we count species. It is a challenging way but it allows to obtain more accurate results. In Figure 4, there are 3 individuals whose 2 are humans and 1 is a dog.



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 species).

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

In order to evaluate our method to count species on camera trap videos, we selected 52 videos. Among these videos, 24 were used to create our own object detection model that we call "train videos", 28 videos correspond to new videos never seen by our model that we call "new videos" whose 4 videos are empty. The "new videos" provided by PNM were not annotated, we manually annotated these videos (species and count) helped by PNM. In both cases the selection of videos was done manually, we tried to select heterogeneous videos which represent all species 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 species (same or different species) with different conditions (species far away or hidden). Thanks to this variety of videos, we can challenge our own model.

4 Experiment and results

4.1 Experiment

4.1.1 Datasets

We have 37,424 single images for 52,470 bounding boxes from day and night for our 13 species. To evaluate the robustness of our model we split these images into 3 datasets: training, validation and test. The train dataset corresponds to 80% of data (29,976 single images), the validation dataset to 10% of data (3,705 single images) and the test dataset to 10% of data (3,743 single images). Since it is possible to have multiple species on same image, the number of single images is different from the number of images by species (Table 2) and the number of bounding boxes by species (Table 1). Furthermore, we considered images are unique in each dataset, they are not entirely randomly affected, an image could not be in different datasets. For instance, if they are more than one species

on an image, the image (with their labels associated) is associated to only one dataset. The aim is to avoid identical images repetitions in multiple datasets in order to not bias our results. In addition, to increase the generalizability of our model with more robust features learned (reduce overfitting) and to generate additional training data, we did data augmentation. We generated new training samples from the original with random horizontal flip. Nevertheless, we faced the problem of unbalanced datasets: the human is overrepresented in comparison to other species.

Table 1: Number of bounding boxes by species in the dataset

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**

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 part, only the 4 last layers (detection and classification biases and weights) are trained to our own 13 species. 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 batch, optimisers, learning rate values and images'sizes. We found optimal parameters thanks to validation dataset. We established optimal parameters are number of batches fixed to 14 and a size of images fixed to 480x270 (original size divided by 4). The best algorithm

Table 2: Number of images by species in the dataset

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**

optimisation is Adam with learning rate value of 1e-5. Moreover, our algorithm can detect 100 bounding boxes on each image, it could be useful in the case of multiple species present at that moment.

Finally, thanks to compare training and validation loss, we selected our best model: 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 datasets. To evaluate performance of our object detection model, for both detection and classification part, 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 admited 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 species 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 dataset are very close to validation dataset, it validates 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 by class can be seen in Figure 6.

Table 3: mAP across validation and test datasets

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%

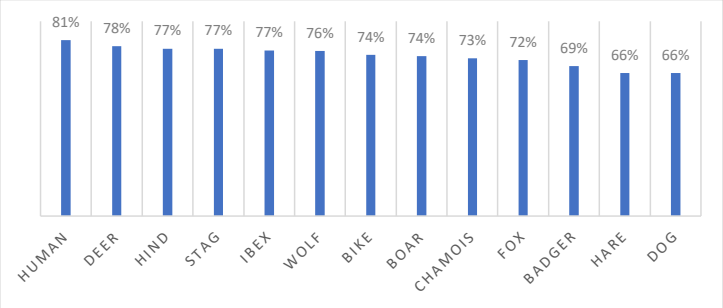
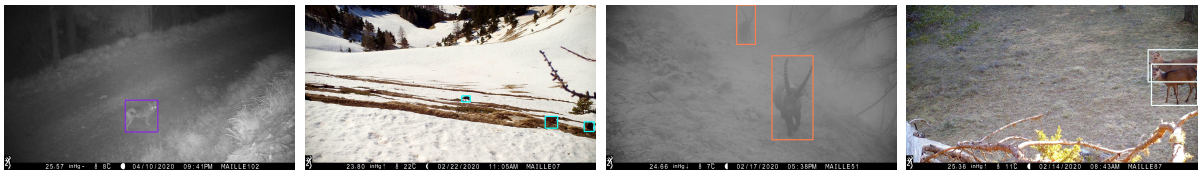


Figure 6: mAP by species across test dataset

Figure 7 shows model capacity to detect species 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 part. As regards to classification part, our model also presents satisfactory results especially for big species and species over-represented like human. As you can see in Figure 8, a good detection and classification of species depends 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: Species detection in different conditions (detection threshold on the BB confidence of 90%)

4.2.2 Count of detected individuals

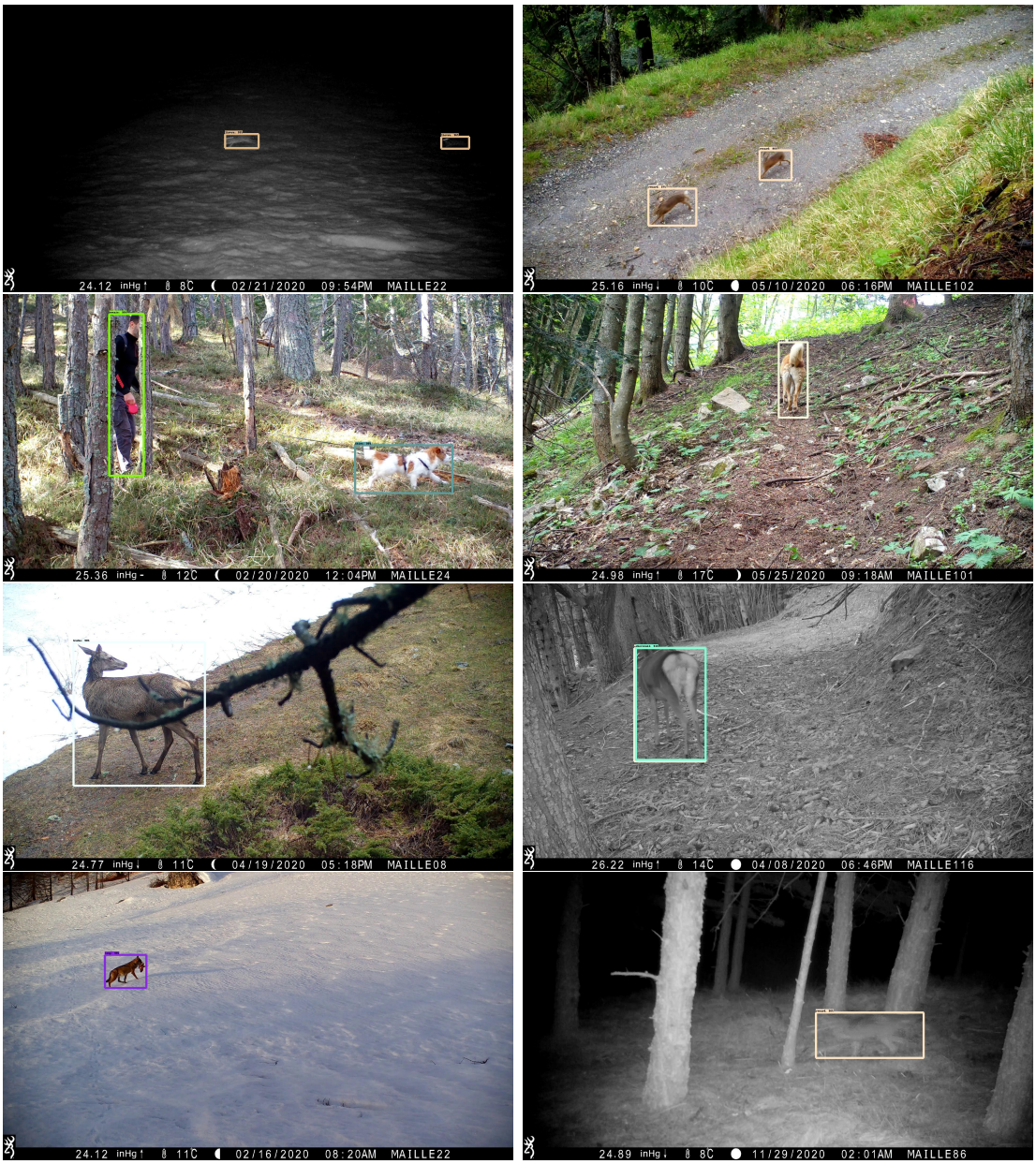
Figure 9 shows results we obtained to predict the number of individual presents on 52 camera trap videos. These results are only based on detection results. Overall, for 62% of videos we predicted the exact number of individuals and for 87% of them the count predicted is either exact or $\pm$ 1 unit. Detection algorithm works well on both dataset 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.

4.2.3 Count of detected species

The results obtained to predict the number of species presents on camera trap videos are showed in Figure 10. Due to the difficulty of the task, to count species based on detection and classification results, lower results are obtained in this part than previous part. We obtained 81 predictions of species associated to count for really 58 species associated to count. Generally, for both "train videos" and "new videos", we predicted 38% rightly species with exactly count and for 48% of videos we rightly predicted species with either exact count or $\pm$ 1 unit. Here, results obtained for "train videos" are better than "new videos". For 55% of "train videos", we predicted the exact number of species whereas for "new videos" is 28%. Finally, concerning empty videos, we reached same results than in previous part, for 50% of them we predicted them correctly to be empty videos. The 50% remaining are predicted with difference of one unit.

5 Discussion

The field of the environment becomes more and more interested in benefiting from recent advances in AI, in particular deep learning methods to avoid to ecologists to do tedious tasks Tuia et al. (2022). In this paper, we demonstrated it is possible to locate, identify



(a) Correct detection and classification (b) Correct detection but misclassified

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

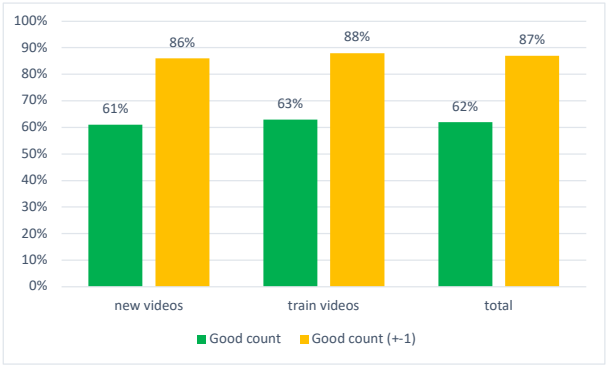


Figure 9: Results of counting individuals detected on camera trap videos

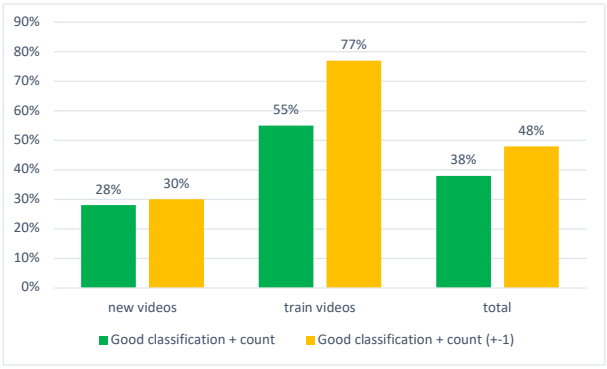


Figure 10: Results of counting species detected on camera trap videos

and estimate population of species on camera trap videos thanks to deep learning models. Firstly, we split videos into images and label them. Then we fine-tune an object detection model for our problem. We obtained convincing results for detection and classification parts. On the one hand, our work demonstrates our 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 cross-validated on the training data. The results on test data show 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 part of our work, our counting method, based on maximum number of bounding boxes detected on one frame, allows to obtain satisfactory results. Our model, DeepWILD, detects and classify 13 species 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 PNM. The model can distinguish empty videos from wildlife or human or vehicles. The performances for detection and classification parts 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 species 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 dataset 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 dataset and to design a model able to detect 13 species 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 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.

Future work As for further directions, we could improve our performance of detection and classification 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. 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 the difficult 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 species provided by PNM (15 other species 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 method, 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, Salazar, Gomez, & Diaz-Pulido, 2019).

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

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

Acknowledgements

This work was made possible thanks to the collaboration with Parc National du Mercantour, which provided 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

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

Beery, S., Liu, Y., Morris, D., Piavis, J., Kapoor, A., Meister, M., ... Perona, P. (2020). Synthetic examples improve generalization for rare classes. In *Proceedings of the ieee/cvf winter conference on applications of computer vision (wacv).*

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

Beery, S., van Horn, G., & Perona, P. (2018). Recognition in terra incognita. In *Proceedings of the european conference on computer vision (eccv).*

Beery, S., Wu, G., Rathod, V., Votel, R., & Huang, J. (2020). Context r-cnn: Long term temporal context for per-camera object detection. In *2020 ieee/cvf conference on computer vision and pattern recognition (cvpr).* doi: https://doi.org/10.1109/CVPR42600.2020.01309

Chen, R., Little, R., Mihaylova, L., Delahay, R., & Cox, R. (2019). Wildlife surveillance using deep learning methods. *Ecology and Evolution*, *9*(17), 9453-9466. doi: https://doi.org/10.1002/ece3.5410

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*

(Vol. 29). Retrieved from https://proceedings.neurips.cc/paper/2016/file/577ef1154f3240ad5b9b413aa7346a1e-Paper.pdf

Ferreira, A. C., Silva, L. R., Renna, F., Brandl, H. B., Renoult, J. P., Farine, D. R., ... Doutrelant, C. (2020). Deep learning-based methods for individual recognition in small birds. *Methods in Ecology and Evolution*, *11*(9), 1072-1085. doi: https://doi.org/10.1111/2041-210X.13436

Giraldo-Zuluaga, J.-H., Salazar, A., Gomez, A., & Diaz-Pulido, A. (2019). Camera-trap images segmentation using multi-layer robust principal component analysis. *The Visual Computer*, *35*(3), 335–347.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 ieee conference on computer vision and pattern recognition.* doi: https://doi.org/10.1109/CVPR.2014.81

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the ieee international conference on computer vision (iccv).*

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 ieee conference on computer vision and pattern recognition (cvpr).* doi: https://doi.org/10.1109/CVPR.2016.90

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

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (Vol. 25). Retrieved from https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

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

Levy, D., Belfer, Y., Osherov, E., Bigal, E., P.Scheinin, A., Nativ, H., ... Treibitz, T. (2018). Automated analysis of marine video with limited data. In *2018 ieee/cvf conference on computer vision and pattern recognition workshops (cvprw).* doi: https://doi.org/10.1109/CVPRW.2018.00187

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the ieee international conference on computer vision (iccv).*

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & C.Berg, A. (2016). Ssd: Single shot multibox detector. In *Computer vision – eccv 2016* (p. 21-37). Springer, Cham. doi: https://doi.org/10.1007/978-3-319-46448-0_2

Norouzzadeh, M. S., Morris, D., Beery, S., Joshi, N., Jojic, N., & Clune, J. (2020). A deep

active learning system for species identification and counting in camera trap images. *Methods in Ecology and Evolution*, *12*(1), 150-161. doi: https://doi.org/10.1111/2041-210X.13504

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packera, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, *115*(25), E5716-E5725. doi: https://doi.org/10.1073/pnas.1719367115

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. In *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr).*

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (Vol. 28). Retrieved from `https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf`

Sarwar, F., Griffin, A., Periasamy, P., Portas, K., & Law, J. (2018). Detecting and counting sheep with a convolutional neural network. In *2018 15th ieee international conference on advanced video and signal based surveillance (avss).* doi: https://doi.org/10.1109/AVSS.2018.8639306

Schneider, S., Taylor, G. W., & Kremer, S. C. (2018). Deep learning object detection methods for ecological camera trap data. In *2018 15th conference on computer and robot vision (crv).* doi: https://doi.org/10.1109/CRV.2018.00052

Shahinfar, S., Meek, P., & Falzona, G. (2020). "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*. doi: https://doi.org/10.1016/j.ecoinf.2020.101085

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

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 31). doi: https://doi.org/10.1609/aaai.v31i1.11231

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In *2015 ieee conference on computer vision and pattern recognition (cvpr).* doi: https://doi.org/10.1109/CVPR.2015.7298594

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

Tuia, D., Kellenberger, B., Beery, S., Costelloe, B. R., Zuffi, S., Risse, B., ... others (2022). Perspectives in machine learning for wildlife conservation. *Nature communications*,

13(1), 1–15.

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

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

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

Willi, M., Pitman, R. T., Cardoso, A. W., Locke, C., Swanson, A., Boyer, A., ... Fortson, L. (2018). Identifying animal species in camera trap images using deep learning and citizen science. *Methods in Ecology and Evolution*, *10*(1), 80-91. doi: https://doi.org/10.1111/2041-210X.13099

Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 ieee international conference on image processing (icip)*. doi: https://doi.org/10.1109/ICIP.2017.8296962

Wäldchen, J., & Mäder, P. (2018). Machine learning for image based species identification. *Methods in Ecology and Evolution*, *9*(11), 2216-2225. doi: https://doi.org/10.1111/2041-210X.13075

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

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

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems* (Vol. 27). Retrieved from `https://proceedings.neurips.cc/paper/2014/file/375c71349b295fbe2dcdca9206f20a06-Paper.pdf`

Yousif, H., Yuan, J., Kays, R., & He, Z. (2019). Animal scanner: Software for classifying humans, animals, and empty frames in camera trap images. *Ecology and Evolution*, *9*(4), 1578-1589. doi: https://doi.org/10.1002/ece3.4747

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

arXiv.1812.04901