Ecological Informatics 41 (2017) 24–32

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks

Alexander Gomez Villa, Augusto Salazar*, Francisco Vargas

Grupo de investigacion SISTEMIC, Facultad de Ingenierıa, Universidad de Antioquia UdeA, Calle 70 No. 52–21, Medellın, Colombia

ARTICLE INFO

Keywords:
Animal species recognition
Deep convolutional neural networks
Camera-trap
Snapshot Serengeti

ABSTRACT

Non-intrusive monitoring of animals in the wild is possible using camera trapping networks. The cameras are triggered by sensors in order to disturb the animals as little as possible. This approach produces a high volume of data (in the order of thousands or millions of images) that demands laborious work to analyze both useless (incorrect detections, which are the most) and useful (images with presence of animals). In this work, we show that as soon as some obstacles are overcome, deep neural networks can cope with the problem of the automated species classification appropriately. As case of study, the most common 26 of 48 species from the Snapshot Serengeti (SSe) dataset were selected and the potential of the Very Deep Convolutional neural networks framework for the species identification task was analyzed. In the worst-case scenario (unbalanced training dataset containing empty images) the method reached 35.4% Top-1 and 60.4% Top-5 accuracy. For the best scenario (balanced dataset, images containing foreground animals only, and manually segmented) the accuracy reached a 88.9% Top-1 and 98.1% Top-5, respectively. To the best of our knowledge, this is the first published attempt on solving the automatic species recognition on the SSe dataset. In addition, a comparison with other approaches on a different dataset was carried out, showing that the architectures used in this work outperformed previous approaches. The limitations of the method, drawbacks, as well as new challenges in automatic camera-trap species classification are widely discussed.

1. Introduction

Automated camera-traps used in wildlife studies are small boxes, secured to a tree, rock or other structure. Camera-traps are powerful tools for wildlife scientists who, thanks to this method, can answer questions such as "Which animal species occurs in a certain area?", "What are they doing?", "How many are there?", among others. Also, fundamental studies, like detecting rare species, delineating species' distributions, documenting predation, monitoring animal behavior, and other vital rates (O'Connell et al., 2010) are supported with this method. Hence, it allows biologists to protect animals and their environment from extinction or man-made damage.

Although camera trapping is a useful method in ecology, this method generates a large volume of images. Therefore, it is a big challenge to process the recorded images and even harder, if the biologists are looking to identify all photographed animals instead of looking for a certain species. Currently, no automatic approach is used to identify species from camera-trap images. Researchers and citizen science volunteers analyze thousands or millions of photographs manually (Swanson et al., 2015). An automatic system that deals with this problem would accelerate the professionals' work, allowing them to focus on data analytics and experimental setup.

Automatic classification of animal species in camera-trap images still remains an unsolved problem due to very challenging image conditions. A few previous works proposed solutions for this problem. Yu et al. (2013) manually cropped and selected images, which contain the whole animal body. This conditioning allowed then to obtain 82% accuracy classifying 18 animal species on their own dataset. Chen et al. (2014) use an automatic segmentation algorithm, but they obtained only 38.3% accuracy.

A publicly available dataset called Snapshot Serengeti dataset (Swanson et al., 2015) was published in 2015. It was acquired with 225 camera-traps placed in Serengeti National Park, Tanzania. There were taken more than one million sets of pictures, each set containing 1–3 photographs. Before the release of the Snapshot Serengeti dataset (Swanson et al., 2015), there was no publicly available dataset to work with and for benchmarking. This dataset allows the computer science community to study and overcome the challenges present in camera trapping framework; it also helps ecology researchers and educators in their camera-trap studies.

* Corresponding author at: SUPSI, App SUPSI-DTI, via Sorengo 22, Lugano 6900, Ticino, Switzerland.
E-mail addresses: alexander.gomezvilla@supsi.ch (A. Gomez Villa), augusto.salazar@udea.edu.co (A. Salazar), jesus.vargas@udea.edu.co (F. Vargas).

http://dx.doi.org/10.1016/j.ecoinf.2017.07.004
Received 17 November 2016; Received in revised form 24 July 2017; Accepted 25 July 2017
Available online 28 July 2017
1574-9541/

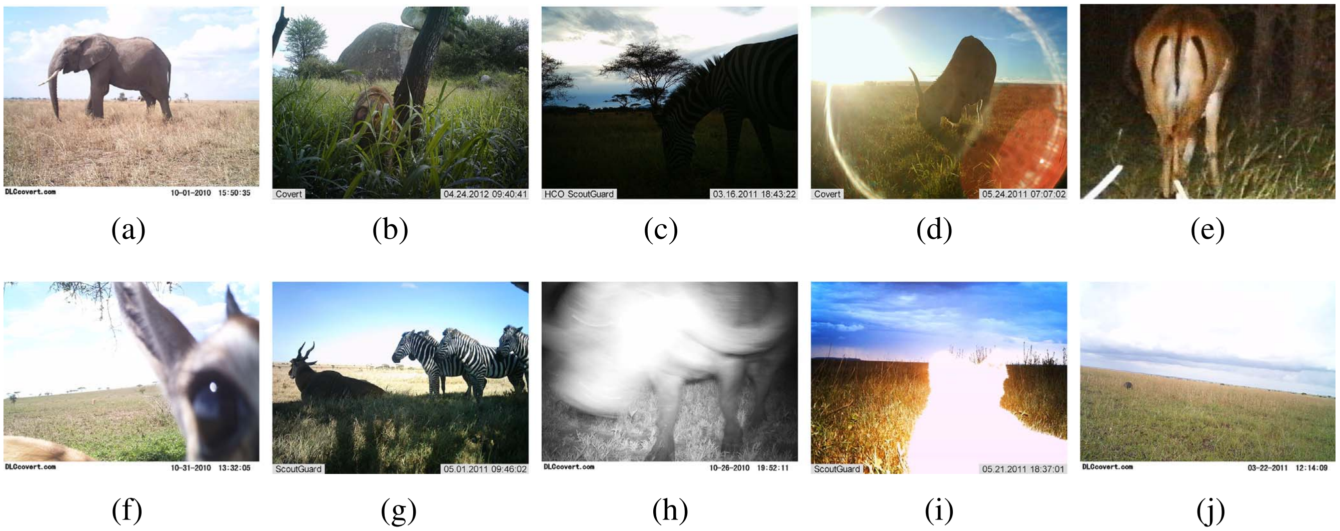

DLCcovert.com 10-01-2010 15:50:35
Covert 04.24.2012 09:40:41
HCO ScoutGuard 03.16.2011 18:43:22
Covert 05.24.2011 07:07:02
DLCcovert.com 10-31-2010 13:32:05
ScoutGuard 05.01.2011 09:46:02
DLCcovert.com 10-26-2010 19:52:11
ScoutGuard 05.21.2011 18:37:01
DLCcovert.com 03-22-2011 12:14:09
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)

Fig. 1. Different camera trapping classification scenarios. (a) Ideal. (b) Occlusion due to context. (c) Poor illumination. (d) Over-exposed regions. (e) Auto-occlusion. (f) Complex animal poses and unexpected images. (g) Different species in the same image. (h) Blurred. (i) Over-exposed animals. (j) Animals far away from camera. All images were take from Snapshot Serengeti dataset (Swanson et al., 2015) as well as all photo-trap images used in this document.

In this work, the main issues inherent to camera trapping images automatic species identification are stated. Through several experiments the capacity of very deep convolutional neural networks to automatize species classification in camera-trap images is proved. Unbalanced, balanced, foreground objects selection, and segmented versions of Snapshot Serengeti dataset are used in order to study how a powerful learning algorithm performs in presence of four of the main issues inherent to camera trapping acquired data: unbalanced samples, empty frames, incomplete animal images, and objects too far from focal distance. The advantages, working conditions and limitations of the approach are explained. Also, a comparison in another dataset with different conditions is done. This comparison shows the flexibility of the model since hardware (cameras), place (Africa–Central America), and species are different. Additionally, this work exposes new challenges and problems in camera-trap automatic species recognition tasks, such as intra-species appearance variance (cubs, male and female) or influence of ambient light in classification. All datasets derived from Snapshot Serengeti used in this work (including manually segmented images) and the trained models, are publicly available.

The rest of the paper is organized as follows: First, related work is discussed in Section 2. In Section 3, the challenges present in camera trapping framework are described. Also, the methods used in the identification model are explained. Section 4 describes the experiments used to test the models. Results are presented in Section 5. Section 6 discusses the results. Finally, in Section 7 conclusions and future work are presented.

2. Related work

This section reviews previous approaches to identify multiple species in camera-trap images. To the best of our knowledge there are only two previous approaches to identify animal species in camera-trap images. Sparse coding spatial pyramid matching (ScSPM) was used by Yu et al. (2013) to recognize 18 species of animals, reaching 82% accuracy on their own dataset (composed of 7196 images). The ScSPM extract dense SIFT descriptors and cell-structured local binary patterns as local features; then global features are generated using global weighted sparse coding and max pooling through multi-scale pyramid kernel. The images are classified using a linear support vector machine. As input to the ScSPM photo-trap images were preprocessed: Removing empty frames (images without animals), manually cropping all the animals from the images, and selecting only those images that contain the animal's whole body (e.g. legs or heads only images are removed).

A deep convolutional neural network (ConvNet) was used by Chen et al. (2014) to classify 20 animal species in their own dataset. An important difference from Yu et al. (2013) is that they use an automatic segmentation method (Ensemble Video Object Cut) for cropping the animals from the images and use this crops to train and test their system. The ConvNet used has only 6 layers (3 convolutional layers and 3 max pooling layers) which give them a 38.31% accuracy.

Our approach uses ConvNets as Chen et al. (2014) but is different in two main aspects. First, an analysis using an unbalanced, balanced, conditioned, and segmented dataset was done. Very deep ConvNets (AlexNet: Krizhevsky et al., 2012, VGGNet: Simonyan and Zisserman, 2014, GoogLenet: Szegedy et al., 2015 and ResNets: He et al., 2015) were used in order to test how a higher learning capacity model reacts to camera-trap automatizing obstacles. In addition, our manually segmented images differ from the work of Yu et al. (2013) in the sense that their crops contain the whole animal body. In contrast, our crops have images containing only some body parts (such as legs and horns), this issue adds high complexity to the classification task (as will be explained later).

3. Towards animal monitoring in the wild

In this section, different situations are described and analyzed that are present in camera-trap images and must be overcome in order to automatically identify species. Also, a solution based on very deep convolutional neural networks is proposed.

3.1. Challenges in camera trapping

Recognition of animal species in camera-trap images can be interpreted as an object recognition problem. In this case, for instance an elephant is present in the image and it must be localized and classified as elephant (see Fig. 1 (a)). Notice that images containing and showing the whole animal body (such as Fig. 1 (a)) are an ideal and scarce case. In camera trapping framework the challenges can be classified in three main groups: Environmental conditions, animal behavior-related, and hardware limitations.

Environmental conditions denotes how the context affects the quality of a camera-trap image. Since the camera-traps are set in the wild and remain there for long periods, many objects can occlude animals, as Fig. 1 (b) shows. As the environment does not remain equal (e.g., plants grow, trees fall, among others), occlusion can appear at any moment. Day and night have different illumination conditions but the transition between them also causes problems (see Fig. 1 (c)). Fig. 1 (d) shows an example of overexposed regions caused by the sun light. Variations like rain and drops on the lens are also examples of conditions that directly affect hardware performance.

A common approach in camera trapping framework is to place the camera pointing towards a natural path or place where the animals are expected to pass through. However, animals hardly behave in a predictable way. A proof of this statement is that very few images contain animals in an ideal condition. A vast majority of camera-trap images do not contain the whole body of an animal due to: Context occlusion, animals too close to the camera (see Fig. 1 (f)), and animal movement. Also, animals are captured in random poses as Fig. 1 (e) and (f) show, which hide important features to recognize a species, and reduce confidence in a classification decision. Particularly in similar species, this fact is crucial, as we will discuss later. Finally, it is also possible that several species appear in one single image (see Fig. 1 (g)) or there is no animal at all in the image (see Fig. 1 (j)). This adds high complexity to the recognition problems, because it forces to localize all animals in the image. The above mentioned situations could happen independently or simultaneously in all possible combinations.

Camera trapping hardware assembles high resistant and low power consumption electronics. There are several models commercially available, which allow different parameter configurations. Resolution of the camera, trigger sensor, frames per second, and night illumination (infrared or flash) are some of the main factors when selecting a camera. Blurred images and overexposed animal images, as Fig. 1 (h) and (i) shows, depend on camera-trap hardware and selected parameters (time between frames and flash power for instance). These problems make shape and fur patterns indistinguishable, which hinders the species identification.

Despite image condition, resembling species recognition turns on the problem in a fine-grained classification task. In Fig. 2 (a) to (e), five similar species found in the same area are shown. Partial (mainly because of inter-frame time or animals close to the camera) and not expected poses can make fine-grained classification even harder (see Fig. 2 (f) and (g)). Also, there is intra-species appearance variation in the same species regarding age and sex of the animal.

An ideal automatic species recognition system must deal with all of the above mentioned conditions. Also, it must classify an animal only with partial information as Fig. 2 (h) shows. Very deep convolutional neural networks are the state of the art in image recognition and are known for their high learning capacity (able to recognize up to 1000 different objects; He et al., 2015) and robustness to typical object recognition challenges. The next section describes the use of convolutional neural networks to solve the species recognition problem.

3.2. Convolutional neural networks (ConvNets)

ConvNets (LeCun et al., 1998) consist of stacked convolutional and pooling layers ending in a fully connected layer with a feature vector shown as output (see Fig. 3). Convolutional layers generate feature maps followed by a non-linear activation function. Pooling layers provide scale invariant capacity to the extracted features. A common topology in a ConvNet consists of many sequential stacked convolutional and pooling layers that can extract discriminative features from an input image.

A transformation that maps from low level to high level features is done in a ConvNet. The first layers contain low level features (e.g., edges and orientation) and the last layers contain high level representation features, such as wrinkles, or in the case of animals, the fur details and its discriminative patterns. An important issue in ConvNets architectures is the Depth (number of layers), reason why the community attempts to boost topology depth. In this work the following mainstream deep architectures are used: AlexNet > : (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2014), GoogLenet (Szegedy et al., 2015), and ResNets (He et al., 2015).

AlexNet is a well known model of ConvNet due to its success in the ImageNet classification challenge (Krizhevsky et al., 2012). It consists of 5 convolutional layers, 3 fully connected layers, and a dropout step in training to deal with overfitting. In the search for more discriminative models, the VGGNet was created. It reached 16 and 19 layers of depth using very small convolutional filters in all convolutional layers. With the same purpose, GoogLenet, in addition to the convolutional and pooling layers, includes the inception module. This new module

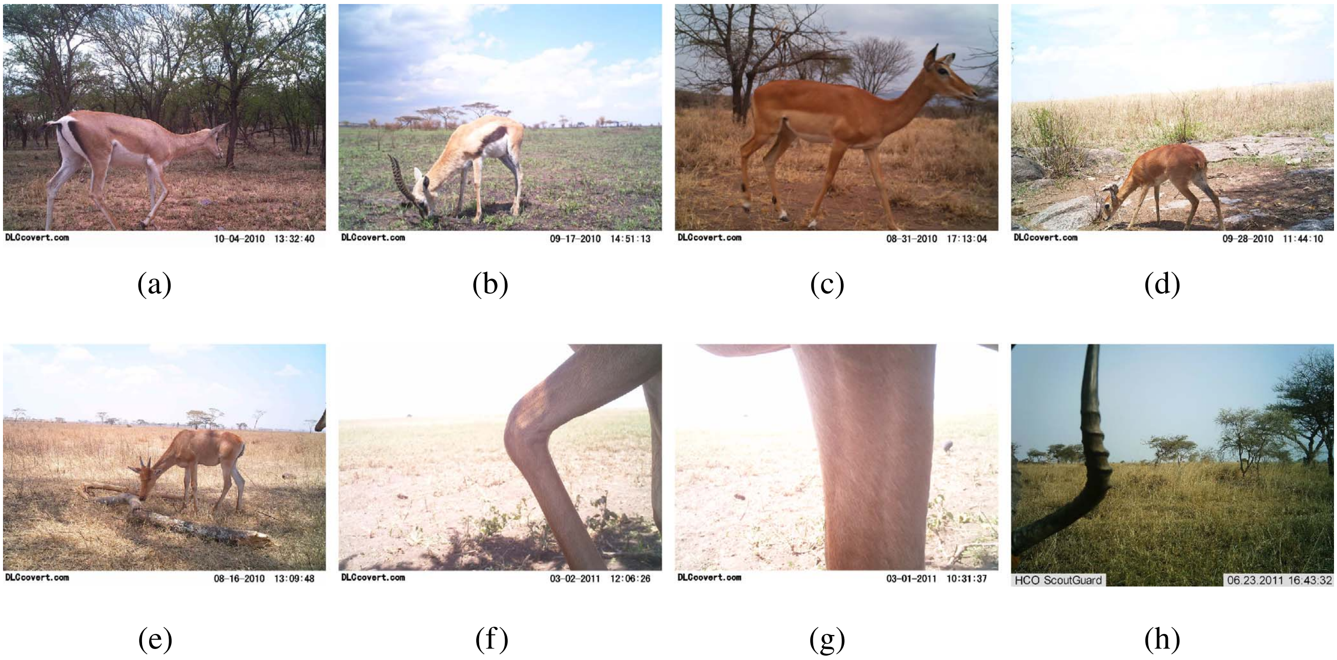

DLCcovert.com 10-04-2010 13:32:40
(a)
DLCcovert.com 09-17-2010 14:51:13
(b)
DLCcovert.com 08-31-2010 17:13:04
(c)
DLCcovert.com 09-28-2010 11:44:10
(d)
DLCcovert.com 08-16-2010 13:09:48
(e)
DLCcovert.com 03-02-2011 12:06:26
(f)
DLCcovert.com 03-01-2011 10:31:37
(g)
HCO ScoutGuard 06.23.2011 16:43:32
(h)

Fig. 2. Similar species in Snapshot Serengeti dataset. (a) Grant's gazelle. (b) Thomson's gazelle. (c) Impala. (d) Reedbuck. (e) Juvenile eland (f) Impala's leg. (g) Impala's leg. (f) Impala's horn.

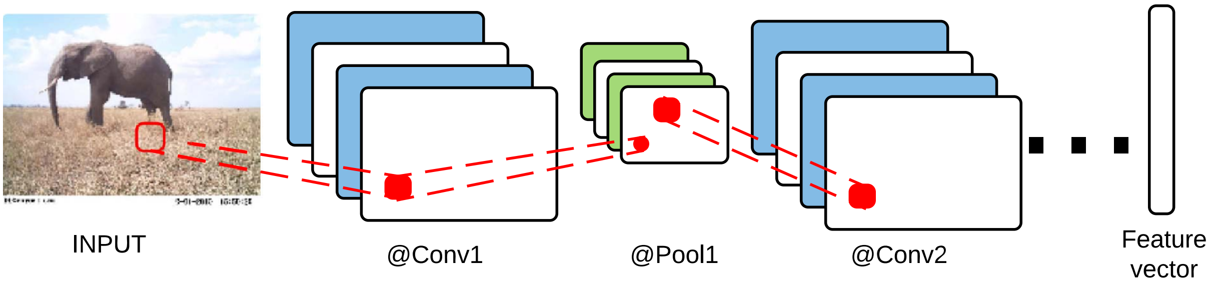


Fig. 3. Convolutional neural network architecture.

extracts non-linear feature maps and adds a sparse connection concept to deal with network size and allows GoogLenet to reach 22 layers. Finally, residual networks address the vanishing gradient problem (a common issue when the Depth increases in deep ConvNets, Glorot and Bengio, 2010) introducing a deep residual learning framework that adds short-cut connections between layers allowing ConvNet to have up to 1000 layers of depth.

In this work, ConvNets are used as black-box feature extractors (using off-the-shelf features, Razavian et al., 2014). Also, the ConvNets are fine-tuned to fit our classification problem. The black-box feature extractor consists of the use of features from highest layers of a ConvNet as inputs of a linear classifier without modifying or training the ConvNet again. This approach is especially useful when the ConvNet was trained with images similar to the new classification problem. A trained ConvNet can be fine-tuned, which means to run the back-propagation algorithm in the pre-trained version of the ConvNet. It is expected that the previous knowledge (contained in the pre-trained ConvNet) helps to improve the learning of the new classification problem. Black box feature extractor and fine-tune process involve a concept called transfer learning (Pan and Yang, 2010).

4. Experimental framework

In this section both, the datasets used and the experiments carried out in this work, are described. Additionally, an explanation of implementation details (such as libraries and architecture parameters) is included.

4.1. Datasets

The Snapshot Serengeti dataset (Swanson et al., 2015) is a camera-trap database published in 2015. It was acquired with 225 camera-traps placed in Serengeti National Park, Tanzania. There were taken 1.2 million image sets of pictures, each set containing 1–3 photographs taken in a single burst per trigger of one second. A particularity of Snapshot Serengeti is that persons from general public (Citizen scientists) annotated the whole sets of images. A consensus classification from multiple annotations was created for each image set as the mode of individual classifications. The Consensus dataset consists of all image sets and their consensus classifications. A subset of this dataset, Gold Standard, was manually annotated by experts.

The Consensus set was validated against Gold Standard revealing 96.6% accuracy for species identification. This is a successful example of citizen science application. In this work a part of the consensus set is used to train and validate the models.

The Snapshot Serengeti contains images of 48 animal species. It is a highly unbalanced dataset, e.g., zebra class has 179,683 images and the striped polecat (zorilla) only 29. In this work, only 26 classes were selected for classification (they are listed in Table 1). The 26 classes were selected using a criteria of quantity: A class must have more than 1000 images (note that in this point we account images not image sets) since it is a common base number in object recognition datasets such as Imagenet (Russakovsky et al., 2015).

Table 1
Set of species selected from Snapshot Serengeti dataset for this work.

Species	# images	Species	# images
Wildebeest	212,973	Lion female & cub	8773
Zebra	181,043	Eland	7395
Thomson's gazelle	116,421	Topi	6247
Buffalo	34,684	Baboon	4618
Human	26,557	Reedbuck	4141
Elephant	25,294	Dik dik	3364
Guinea fowl	23,023	Cheetah	3354
Giraffe	22,439	Hippopotamus	3231
Impala	22,281	Lion male	2413
Warthog	22,041	Kori Bustard	2042
Grants gazelle	21,340	Ostrich	1945
Hartebeest	15,401	Secretary bird	1302
Spotted hyena	10,242	Jackal	1207

In order to compare with previous approaches, the camera-trap dataset used in Chen et al. (2014) was also used (this dataset will be referenced as Panama dataset in the following). It contains 20 species common to North America and has 14,346 training images and 9530 testing images (the train-test partition was done by Chen et al. (2014)). These images were generated using an automatic segmentation algorithm described in Ren et al. (2013). The segmentation algorithm uses the burst of camera-trap photos to segment the moving objects. The Panama dataset contains color and infrared images. This dataset (or at least the one that was shared with the authors of this paper) is that it also contains images without animals that have a species label due to the fact that the segmentation algorithm does not work perfectly. No photos were eliminated or selected.

4.2. Experiments

The dataset of 26 selected classes from the Consensus set were divided into several versions (see Table 2) using a pool of images made of image sets marked as “contain an animal”. Tests using $D1$ dataset are aimed to know how well the models deal with the highly unbalanced nature of camera trapping databases. This dataset is composed of all images listed in Table 1. To examine the impact of unbalanced data, the $D2$ dataset is used (selecting 1240 images per species from $D1$). The aim of this experiment is to see if the models are able to fit easier than in the unbalanced case. This is a subset from $D1$, where 1000 and 240 images per class were randomly selected as train and evaluation sets, respectively. Dataset $D3$ was generated from images where the animal is

Table 2
Snapshot Serengeti dataset partitions. Each dataset version contains 26 classes (species).

Label	Description	Training set	Test set
$D1$	Raw unbalanced images	548,608	235,118
$D2$	Raw balanced (1240 per species) images	26,000	6240
$D3$	Images with animals in foreground	21,630	6240
$D4$	Animals manually segmented from $D3$	22,936	6240

Table 3
Architectures used in the experiments

Label	Architecture	# layers
A	AlexNet	8
B	VGG Net	16
C	GoogLenet	22
D	ResNet-50	50
E	ResNet-101	101
F	ResNet-152	152
G	AlexNet-FT	8
H	GoogLenet-FT	22

placed in the foreground (manually selected from *D*2). These conditions are ideal and difficult to reach in practice. For instance, a sensor with a shorter range could be used, but this is not a desired condition for biologists since they want to capture as many images (with animals) as possible. Finally, dataset *D*4 derived from *D*3, consists of manually segmented images to simulate a segmentation algorithm that always finds the animal (or part of it) in the image. This conditioning differs from Yu et al.'s (2013) cropped images, where the cropped images only contain whole bodies of animals. In our case the cropped images include images where the animals appear partially (not the whole body inside the image e.g. images of legs, horns, heads). When multiple animals were present in the same image, several sub images were manually cropped. Hence, an image with three zebras will result in three images for *D*4. The test set was kept balanced always (discarding some images with several animals per image).

Table 3 shows the six very deep ConvNets (labels A to F) used in this work. They are the state of the art in object recognition. Fine-tuning was not always possible due to our hardware limitations, but the results will show how fine-tuning impacts in two models performance (AlexNet and GoogLenet architectures with labels G and H). This work uses multiple very deep ConvNets in order to prove how the depth in ConvNets impacts on the camera trapping classification problem and to answer questions such as "Could a deeper model deal with empty frames?", "Could a deeper model deal with an unbalanced dataset?", among others. In each dataset partition (*D*1 to *D*4), eight experiments were carried out using the listed architectures in Table 3, which gives a total of 32 experiments.

To compare with previous approaches an experiment labeled as C1 was done. In this experiment, the model (ConvNet architecture) that reached the highest accuracy in the Snapshot Serengeti dataset experiments was trained and tested using the Panama dataset. The training and testing procedure was the same as used for snapshot Serengeti dataset partitions. It is described in Section 4.3.

Accuracy in the validation set is used as performance metric. Top-1 (when the correct class is the most probable class) and Top-5 (when the correct class is within the five most probable classes) accuracies are presented as a way to tell how well the model is ranking possible species, and also to explore automatic species classification as a helper when the number of species increases. This metric is a common choice in recognition tasks like the ImageNet recognition challenge (Russakovsky et al., 2015).

4.3. Implementation details

All the dataset images were re-sized to fit in the ConvNet topologies input: AlexNet (227 × 227), VGGNet (224 × 224), GoogLenet (224 × 224), and ResNets (224 × 224). To use ConvNets as feature extractors the last fully connected layer was modified to deal with 26 classes instead of 1000 Imagenet challenge classes. Hence, all last full connected layers from all topologies were replaced to deal with 26 classes.

All used architectures were pre-trained with the ImageNet dataset (Russakovsky et al., 2015). The fine-tuning process was done running the back-propagation algorithm with Stochastic gradient descent in the pre-trained topologies. Both the learning rate and step size were reduced in order to deal with overfitting and let the network learn slowly. To fine-tune the models the learning rate of each layer was modified by one layer per test. After the performance is calculated, one more layer is allowed to learn and the process is repeated until the performance stops increasing. The process of fine-tuning begins from last to first layer. This is based on Zeiler and Fergus (2014) work, which suggests that the last layers learn more specific class representations. In the pre-trained models, the last layers have a specialization in 1000 ImageNet classes. Ideally, this knowledge will be replaced with 26 snapshot Serengeti classes through the fine-tuning procedure.

The implementation was done in the deep learning framework Caffe (Jia et al., 2014), as well as all pre-trained models that were found in the Caffe model Zoo. The Caffe models, the configuration files and the Snapshot Serengeti dataset version used in this paper (versions *D*1 to *D*4) are publicly available for benchmarking at https://computervisionlabudea.wordpress.com/identification-of-animal-species-on-camera-trap-images/.

5. Results

The results are organized as follows. First, the influence of raw unbalanced data (*D*1) in the classification models is shown. Second, a balanced dataset (*D*2) classification approach is presented. Then, results using *D*3 and *D*4 as input to classification models are exposed. A clarification of the experiments using fine-tuned architectures is done. Finally, the results in the Panama dataset and a comparison against Chen et al. (2014) approach is showed.

Fig. 4 shows the results of experiments using the Snapshot Serengeti dataset. Top-1 and Top-5 results are alongside each experiment, for instance for the experiment with dataset *D*1 and architecture A (first bars of Fig. 4 (a)), the Top-1 and Top-5 reach an accuracy values of 35.4% and 60.4%, respectively.

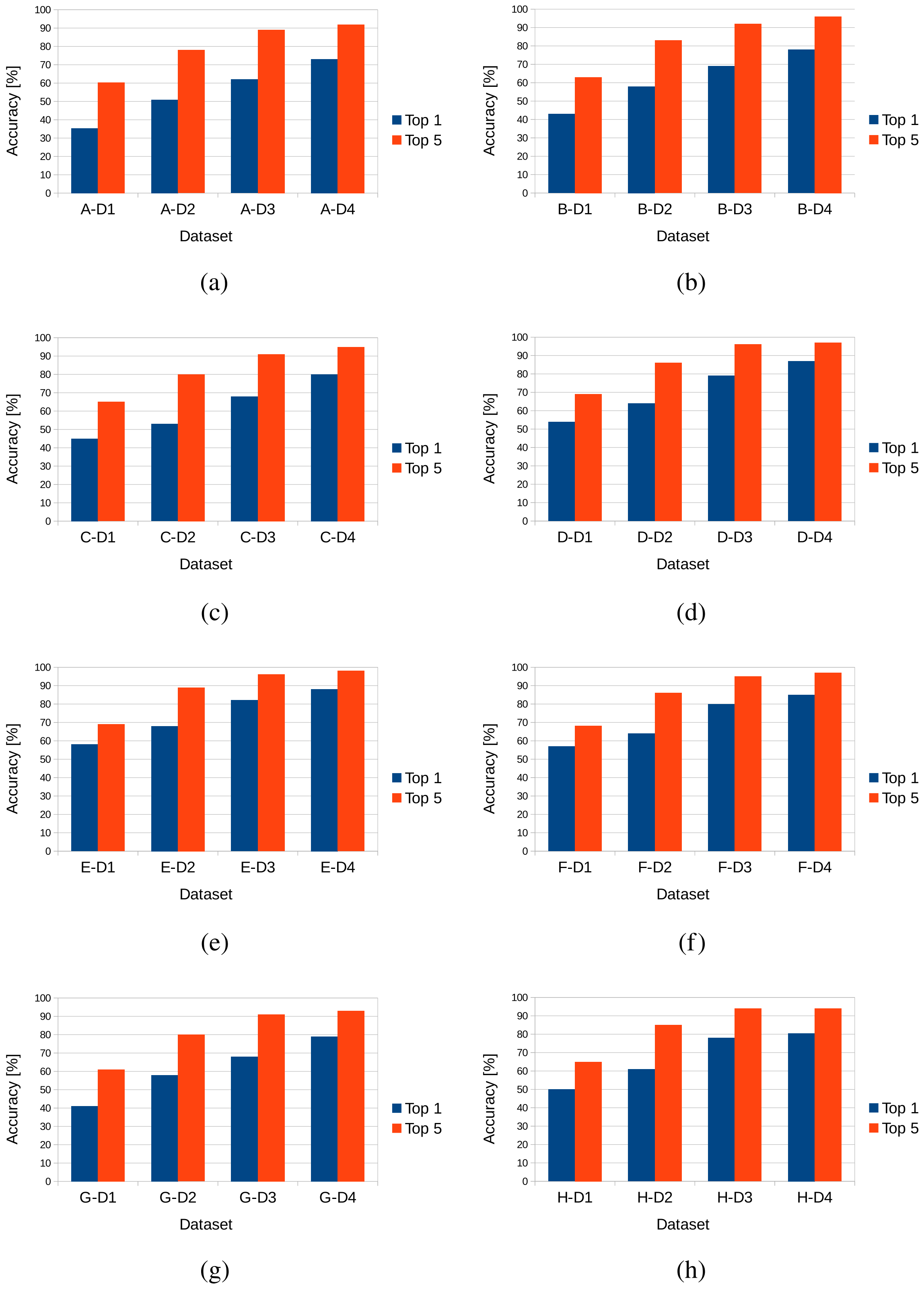
The results with the dataset D1 (see Fig. 4 (a) to (h) first columns) were worst among four dataset partitions. Notice that even the deepest architectures cannot deal with this highly unbalanced data. However, the deepest models outperform less deep ones.

As for experiments with dataset *D*2, the balanced feature is simulated. In this case, the accuracy is higher than for the experiments with dataset *D*1 (see Fig. 4 (a) to (h) second pair of columns). Similar to the unbalanced case, the better performance was obtained with very deep topologies. However, the performance still remains under 70% of accuracy. A cause of this results could be that the dataset includes images of both empty frames (due to in some image sets animals appears in one or two images) and animals far away from the camera. The latter ones become empty frame images when the image is re-sized to fit in the ConvNets. These two issues (interpreted as noise) may be handled by the classifier (we will discuss later) but only if the number of instances is small compared to the number of images with animals in foreground.

Although the problems present in the images of the dataset *D*2 (images without animals) depend on time between shots, the place of the camera, and sensor parameters, which can be controlled, the animal's behavior is still an uncontrolled factor. Therefore, these two issues always will be part of the problem and an automatic identification process must consider these issues.

The results using dataset *D*3 show much better performance than the two previous cases, reaching an accuracy of 82.9% and 96.8% for Top-1 and Top-5, respectively (see Fig. 4 third bar) in architecture E. This results confirm that the empty frames, which were removed, influence the classification ability of the model. Again, the better results were obtained using very deep architectures (Residual networks).

The best results were obtained using the dataset *D*4 and the E architecture (88.9% Top-1 and 98.1% Top-5, see Fig. 4 (a) to (h) third pair of columns). As for the previous cases, the very deep models (Residual networks) produced the better results. This is evidence of the



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4. Results of the experiments in the Snapshot Serengeti dataset versions used in this work. (a) Using architecture A. (b) Using architecture B. (c) Using architecture C. (d) Using architecture D. (e) Using architecture E. (f) Using architecture F. (g) Using architecture G. (h) Using architecture H.

ability of the ConvNets architectures to deal with the species classification problem, even in cases where the evaluated images contain only parts of the animal body. Although the accuracy values with the dataset *D*4 were the highest, the difference with respect to the ones with the dataset *D*3 is not as noticeable as for the first two sets of experiments (using datasets *D*1 and *D*2). This could be an indicator that the segmentation may not be necessary but a deeper model must be tested instead. Note that although architecture F is deeper than E its representation power is not so different (as their similar performance show). Since in this work only fully connected layers were trained

Table 4
Species recognition performance on Snapshot Serengeti balanced dataset of the most common 26 species with only animals in the foreground and the animals manually cropped (D4). A comparison against Citizen science accuracy with respect to experts is shown too. All the accuracies are expressed in percentage [%].

Species	ConvNet accuracy	Citizen science accuracy (Swanson et al., 2015)	Species	ConvNet accuracy	Citizen science accuracy (Swanson et al., 2015)
Guinea fowl	99.5	98.2	Hippopotamus	94.1	100
Zebra	99.5	99.7	Wildebeest	93.1	99.3
Human	99.1	96.0	Jackal	92.4	33.0
Baboon	98.3	100	Impala	92.0	95.0
Warthog	98.0	100	Elephant	90.4	98.8
Kori bustard	97.9	100	Eland	87.5	92.0
Hartebeest	97.5	97.6	Spotted hyena	85.4	100
Cheetah	97.0	100	Buffalo	83.3	96.8
Giraffe	96.6	100	Lion female & cub	83.3	100
Topi	95.8	75.0	Lion male	77.0	100
Secretary bird	95.4	100	Thomson's gazelle	71.6	99.0
Reedbuck	95.0	88.0	Dik dik	75.8	100
Ostrich	94.1	100	Grant's gazelle	65.0	82.1

(which did not put Resnet-152 above Resnet-101 in performance), it is necessary to fine-tune the last convolutional layers of Resnet-152 in order to outperform Resnet-101.

Regarding to the fine-tuned versions of ConvNets, architectures A and C outperformed the black boxes networks in all dataset versions. This result shows how fine-tuning specialized the network on the camera-trap classes and suggests that if deeper architectures are fine-tuned, they probably will outperform our best results. Unfortunately, due to our hardware limitations this hypothesis could not be proved.

Table 4 shows the accuracy results per class using the architecture E trained with dataset *D*4. In most of the classes the accuracy reached high performance. The low performance were in the classes related to the fine-grained classification problem. In twenty classes the citizen's classification accuracy outperformed the automatic approach performance. In some cases (such as Zebra or Hartebeest) the model is near human performance and it easily outperforms humans (between citizens and automatic approach) in species with a "Deer" appearance, but without or lacking characteristic pattern. (such as Topi and Reedbuck).

Table 5 presents the results of experiment C1. The model used in this experiments was the ResNet-101 (architecture E). In most of the cases, the results were better, except for Red Deer and Wood Mouse. These results show how a deeper architecture outperforms the less deeper one used in Chen et al. (2014). Note that in this work the ConvNet was fine-tuned (hence the feature extractor is already trained). As was stated before, this proves that the automation of camera-trap species recognition depends not only on having enough data but also on having a powerful learning algorithm and good initialization in the ConvNet.

Table 5
Comparison of species recognition performance on camera-trap dataset (Chen et al., 2014) using very deep convolutional neural networks. The highest accuracy is bold in each species.

Species	ConvNet (Chen et al., 2014)	ResNet 101	Species	ConvNet (Chen et al., 2014)	ResNet 101
Agouti	13	**51.2**	Tinamou	29.8	**59.2**
Peccary	12.2	**69.6**	W-tail deer	50.0	**63.3**
Paca	18.7	**45.2**	Mouflon	71.0	**87.2**
R-brocket deer	2.0	**13.8**	R-deer	**82.0**	60.7
W-nosed coati	24.3	**63.0**	Roe deer	4.6	**50.8**
Spiny rat	5.0	**9.0**	Wild boar	17.1	**44.3**
Ocelot	22.4	**59.2**	R-fox	1.0	**6.4**
R-squirrel	3.8	**32.8**	Euro hare	2.0	**6.3**
Opossum	14.7	**41.6**	Wood mouse	**87.3**	86.9
Bird spec	0.1	**29.8**	Coiban agouti	4.5	**53.8**

6. Discussion

In this work, models (ConvNets) that can classify 1000 classes with a 19.38% of error (as was reported in He et al., 2015) are used. But even more conditioned versions of the dataset could not reach an accuracy higher than 88.9%. If our task is a 26 classes problem, which in some sense is simpler than the 1000 ImageNet identification challenge, what is the big deal? Fig. 5 shows some classification errors in the evaluation set. From Fig. 5 (a) to (f), just a part of the animal body appears. In this cases, the models do not have enough information to predict the correct class. This error shows how the ConvNet is specialized in certain parts of animals (e.g black line in Grant's gazelle Fig. 5 (f)) to recognize the species. For instance, when a "deer" appearance animal (such as Grant's Gazelle, Thomson's Gazelle, Impala, or Reedbuck) with a black line image is evaluated in the model, its output will be with high probability "Thomson's Gazelle". However, if the same image does not have a black line the prediction is less secure, and the model tends to fail, if there is not another species characteristic pattern (such as horns, fur pattern, among others).

In Fig. 5 (e), the fine-grained problem previously explained appears. The ConvNet assumes a black line in the body as a feature of Thomson's Gazelle but this is not a deterministic difference between Grant's and Thomson's Gazelle (this black line can also appear in Grant's Gazelle, perhaps it is rare). In Fig. 5 (f), the fine-grained classification is even harder, due to the fact that this back pattern is similar in Grant's Gazelle and impala species.

The case shown in Fig. 5 (g) exhibits how the ConvNet has to learn more specific features to correctly predict classes in poses that do not show characteristic patterns. In an extreme case the ConvNet has to take a low trusted decision if the visual information is not enough. There are a lot of images that are in previously mentioned conditions. Top-5 accuracy in Fig. 4 shows a high accuracy, which evidences that even if the prediction was wrong the ConvNet is near the correct answer and learned a correct and discriminative pattern for each species.

Fig. 5 (d) shows a buffalo calf. The appearance of the species can be different in young age. The cub of a buffalo does not have the distinctive pattern learned by the model (for buffalo class). This issue arises the necessity of samples from cubs, female, and male individuals of the same species.

In addition, the fact that Fig. 5 (h) was classified as hyena is related to two issues. First, the shape of the animal species (buffalo cub as hyena or buffalo as hippopotamus) could resemble. Second, the image was taken in gray scale, mostly all images from hyenas are in gray scale (hyenas are used to be captured at night). This issue bias the ConvNet to classify all gray scale images with a "Hyena" shape as hyena or

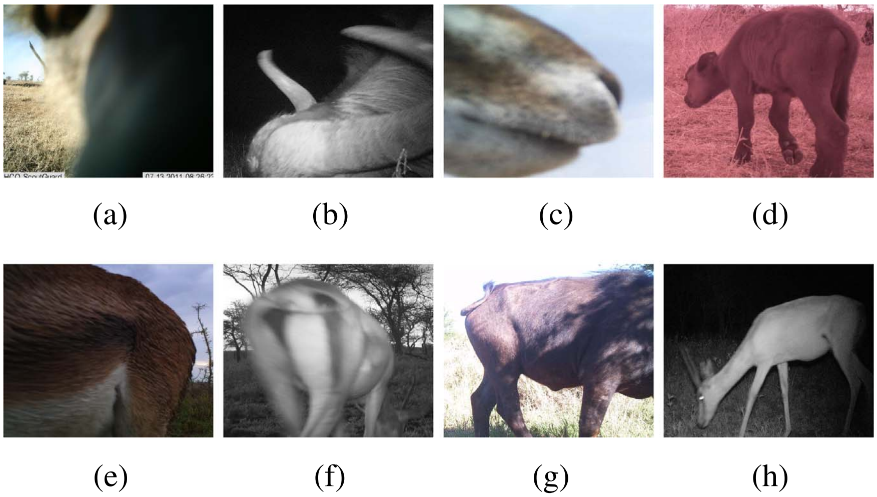


Fig. 5. Misclassified images. (a) Baboon classified as impala. (b) Buffalo classified as wildebeest. (c) Grant's Gazelle classified as Giraffe. (d) Buffalo classified as Hyena. (e) Grant's Gazelle classified as Thomson's Gazelle. (f) Grant's Gazelle classified as Impala. (g) Buffalo classified as Wildebeest. (h) Grant's Gazelle classified as Hyena.

"Hippopotamus" shape as hippopotamus. To deal with this problem the Hyena class must be provided with more color images. Additionally, meta-data (such as hour, date, and location of the image) can be used to deal with the day-night bias problem giving this information to the classifier.

The Snapshot Serengeti dataset was annotated by citizens and experts. Although the comparison between experts and amateurs gives high accuracy, some mistakes occurred. In Van Horn et al. (2015), it is discussed how learning algorithms are robust to annotation errors and training data corruption. This assumption is true only if the training set has sufficient samples. In this work, another citizen annotated dataset was used and the results show robustness to data corruption. Hence, these results confirm the discussion of Van Horn et al. (2015) and motivate the use of citizen science. How much impact data corruption has on training, evaluation, and how big the dataset has to be to deal with this annotation errors, are important questions that must be answered.

In this work, the most common 26 species from Snapshot Serengeti are used, but the dataset has 48 animal species. This means that 28 classes can not be reliable classified due to the low number of samples. However, these rare and scarce species are critical to biologists. This issue can be tackled in several ways (assuming more samples cannot be recorded):

- An extra class called "others" can be added merging the remaining 28 classes. Additionally, a confidence threshold metric, which accepts or rejects the automatic model classification, must be used, too. The confidence threshold will avoid bigger classes to absorbing samples from scarce ones at the price of human work (please see Gomez et al. (2016) for details).
- A set of one-class classifiers (one for each species) using features from a pre-trained ConvNet can be used. As in the first point mentioned, the confidence of the decision must be carefully monitored, in order to avoid false positives in bigger classes.
- Training the ConvNets with the full dataset (48 classes) and punishing the model during training, if it makes a prediction mistake in low sample classes, is another approach. Success is not guaranteed with this approach, but it is a common method to tackle unbalanced classes.
- A further approach would be using one-shot learning in the whole dataset. Note that this approach has not been used in camera-trap images yet (to the best of the authors knowledge)

7. Conclusions

In this paper, an automatic species recognition method based on very deep convolutional neural networks is proposed. Extensive experiments using four different subsets from the Snapshot Serengeti dataset (that reflects possible automatized scenarios for camera-trap framework) and six state of the art deep convolutional neural networks, were carried out. Our method achieves 88.9% Top-1 accuracy and 98.1% Top-5 accuracy in a balanced dataset version of the most common 26 species (with animals in the foreground and manually cropped). The results show to which extend learning algorithms are robust to not perfectly annotated data by citizens (consensus set), bad segmented images, and empty images. Also, multiple problems in the classification task source of low performance are depicted, such as unbalanced datasets, animals far from the camera, animals very close to the camera, segmentation difficulties, animals of different age and sex classes, black-and-white versus color images, and images without the presence of animals. The experiments exhibit that it is possible to obtain highly accurate classification of species from camera-trap images, if there is sufficient data and an accurate segmentation algorithm (exploiting a trade-off between data quantity, model learning capacity and segmentation algorithm performance) available. The versions of the Snapshot Serengeti dataset, as well as manually segmented images, and deep architecture models are publicly available for benchmarking and make the use of the Snapshot Serengeti dataset for computer vision research easier. In addition, a comparison with a previous approach (Chen et al., 2014) was drawn. The presented method outperforms the previous results in most of the cases.

In order to fully automatize the camera-trap method the following issues (ordered by priority) must be addressed:

- Development of a segmentation algorithm that approximates to manual segmentation. This ideal algorithm will receive a full record of camera-trap images and it will output segmentation of animals present in the data (discarding all empty images in the process).
- Incorporation of a classification based on whole sets of images instead of single image classification.
- Increase of dataset size in a smart way. Which means include same number of samples per class and be sure to have all possible intra-class variation modes (cubs, female, male, night, day, poses and parts of the animals). Strategies such as data augmentation and transfer learning must be used as an option to lessen the unbalanced problem.

Our short-term goals are to develop a segmentation algorithm that approximates to manual segmentation in order to alleviate the classifier's data dependence and to guarantee high performance in classification. This is a high priority task in order to fully automatize camera-trap methods. Once better segmentation algorithms will be available, it could be possible to develop a software to assist biologists in the camera-trap field.

In the future, the system will be tested using temporal information. Since camera trapping framework usually offers a burst of images (currently, each image is analyzed independently), most of the problems explained in Section 3 could be solved, if the sequence of images is evaluated as a single instance (as biologists do). New datasets will be produced, which will be split in sets corresponding the problems mentioned in Fig. 1:

- Ideal
- Occlusion due to context
- Poor illumination
- Over-exposed regions
- Auto-occlusion
- Complex animal poses and unexpected images
- Animals far away from camera

Evaluating the classification models in these datasets will allow quantifying and understanding the error introduced by these problems in the automatic approach.

Since the classification of camera-trap images requires fine-grained classification, more training images using parts of animals (cropping existing *D*4 images) will be included into the models. Using ImageNet cropped images (with animal parts) will increase the training set and probably increase the performance.

References

Chen, G., Han, T.X., He, Z., Kays, R., Forrester, T., 2014. Deep convolutional neural network based species recognition for wild animal monitoring. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 858–862.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: International conference on artificial intelligence and statistics, pp. 249–256.

Gomez, A., Diez, G., Salazar, A., Diaz, A., 2016. Animal identification in low quality camera-trap images using very deep convolutional neural networks and confidence thresholds. In: International Symposium on Visual Computing, pp. 747–756.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

O'Connell, A.F., Nichols, J.D., Karanth, K.U., 2010. Camera Traps in Animal Ecology: Methods and Analyses. Springer Science & Business Media.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S., 2014. CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813.

Ren, X., Han, T., He, Z., 2013. Ensemble video object cut in highly dynamic scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1947–1954.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) 115 (3), 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-scale Image Recognition. arXiv preprint arXiv:1409.1556.

Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., Packer, C., 2015. Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. Sci. data 2.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S., 2015. Building a bird recognition app and large scale dataset with citizen scientists: the fine print in fine-grained dataset collection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 595–604.

Yu, X., Wang, J., Kays, R., Jansen, P.A., Wang, T., Huang, T., 2013. Automated identification of animal species in camera trap images. EURASIP J. Image Video Process. 2013 (1), 1–10.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: Computer Vision-ECCV 2014. Springer, pp. 818–833.