2017 International Conference on Data Science and Advanced Analytics

Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring

Hung Nguyen[1], Sarah J. Maclagan[2], Tu Dinh Nguyen[1], Thin Nguyen[1],
Paul Flemons[3], Kylie Andrews[4], Euan G. Ritchie[2] and Dinh Phung[1]

[1]*Deakin University, Geelong, Australia, Centre for Pattern Recognition and Data Analytics*
[2]*Deakin University, Burwood, Australia, Centre for Integrative Ecology*
[3]*Australian Museum Research Institute, Sydney, Australia*
[4]*ABC Radio National, Australia*
{hung, smaclaga, tu.nguyen, thin.nguyen}@deakin.edu.au,
paul.flemons@austmus.gov.au, andrews.kylie@abc.net.au, {e.ritchie, dinh.phung}@deakin.edu.au

***Abstract*—Efficient and reliable monitoring of wild animals in their natural habitats is essential to inform conservation and management decisions. Automatic covert cameras or "camera traps" are being an increasingly popular tool for wildlife monitoring due to their effectiveness and reliability in collecting data of wildlife unobtrusively, continuously and in large volume. However, processing such a large volume of images and videos captured from camera traps manually is extremely expensive, time-consuming and also monotonous. This presents a major obstacle to scientists and ecologists to monitor wildlife in an open environment. Leveraging on recent advances in deep learning techniques in computer vision, we propose in this paper a framework to build automated animal recognition in the wild, aiming at an automated wildlife monitoring system. In particular, we use a single-labeled dataset from Wildlife Spotter project, done by citizen scientists, and the state-of-the-art deep convolutional neural network architectures, to train a computational system capable of filtering animal images and identifying species automatically. Our experimental results achieved an accuracy at 96.6% for the task of detecting images containing animal, and 90.4% for identifying the three most common species among the set of images of wild animals taken in South-central Victoria, Australia, demonstrating the feasibility of building fully automated wildlife observation. This, in turn, can therefore speed up research findings, construct more efficient citizen science-based monitoring systems and subsequent management decisions, having the potential to make significant impacts to the world of ecology and trap camera images analysis.**

***Index Terms*—deep learning, convolutional neural networks, large scale image classification, animal recognition, wildlife monitoring, citizen science**

I. Introduction

Observing wild animals in their natural environments is a central task in ecology. The fast growth of human population and the endless pursuit of economic development are making over-exploitation of natural resources, causing rapid, novel and substantial changes to Earth's ecosystems. An increasing area of land surface has been transformed by human action, altering wildlife population, habitat and behavior. More seriously,

(a) a Reconyx covert camera (b) a camera trap deployed in the wild

Figure 1: An example of camera trap setting in the open space (source: http://reconyx.com.au/gallery.php, 08 June 2017).

many wild species on Earth have been driven to extinction, and many species are introduced into new areas where they can disrupt both natural and human systems [1]. Monitoring wild animals, therefore, is essential as it provides researchers evidences to inform conservation and management decisions to maintain diverse, balanced and sustainable ecosystems in the face of those changes.

Various modern technologies have been developed for wild animal monitoring, including radio tracking [2], wireless sensor network tracking [3], satellite and global positioning system (GPS) tracking [4], [5], and monitoring by motion-sensitive camera traps [6]. Motion-triggered remote cameras or "camera traps" are an increasingly popular tool for wildlife monitoring, due to their novel features equipped, wider commercial availability, and the ease of deployment and operation. For instance, a typical covert camera model (Figure 1) is capable of not only capturing high definition images in both day and night, but also collecting information of time, temperature and moon phase integrated in image data. In addition, generous and flexible camera settings allow tracking animals secretly and continuously. Once being fully charged, a

978-1-5090-5004-8/17 $31.00 © 2017 IEEE
DOI 10.1109/DSAA.2017.31

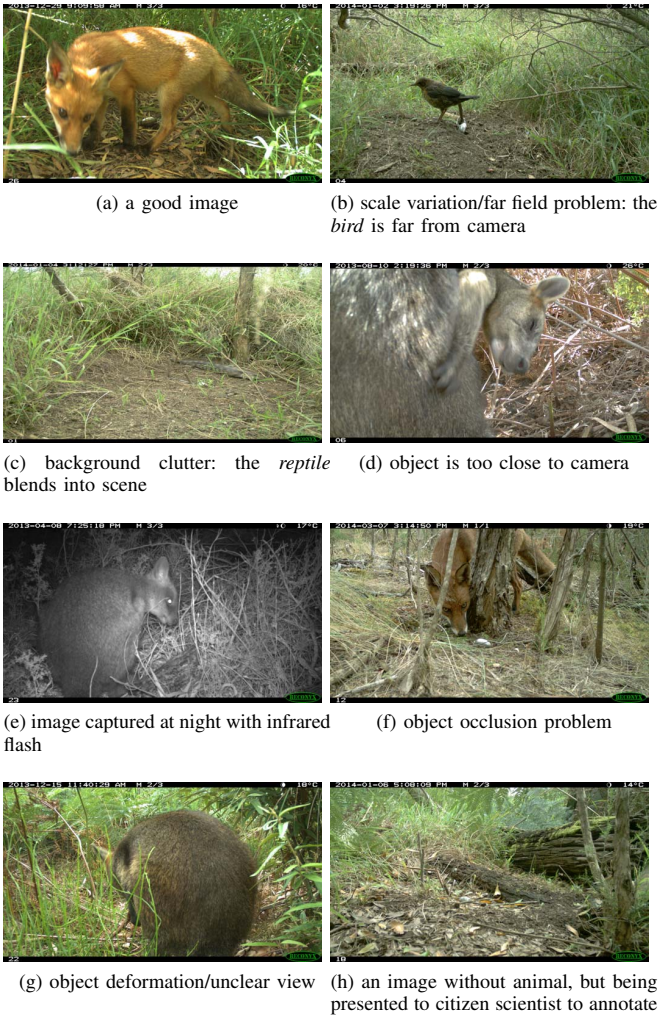
camera can snap thousands of consecutive images, providing a large volume of data. These specifications make camera traps a powerful tool for ecologists as they can document every aspect of wildlife [7].

Visual data, if can be captured, is a rich source of information that provide scientists evidences to answer ecology-related scientific questions such as: what are the spatial distributions of rare animals, which species are being threatened and need protection such as *bandicoot*, which cohort of pest species, such as *red fox* and *rabbit*, need to be controlled; these are examples of key questions to understand wild animals' populations, ecological relationships and population dynamics [7]. To this end, a recently widely-used approach by ecologists is to set up several camera traps in the wild to collect image data of wild animals in their natural habitats [6], [7], [8].

Camera trapping is rapidly being adopted for wildlife monitoring thanks to advances in digital technology that produce more modern camera traps with automation of system components but lower cost of purchase; the task of analyzing huge collections of camera trap images, however, has been conducted manually. Despite the fact that human visual system can process images effortlessly and rapidly [9], processing such an enormous number of images manually is much expensive. For example, to date, the Snapshot Serengeti project[1] gathered 3.2 million images through 225 camera traps across the Serengeti National Park, Tanzania from 2010–2013 [8]. Another similar project, Wildlife Spotter[2], collected millions photos of wildlife captured in tropical rainforests and dry rangelands of Australia. Unfortunately, due to automatic trap camera snapping mechanism, the vast majority of captured images are challenging to process, even for human. Only a limited number of collected images are in favorable condition as in Figure 2a. Many images contain only partial body of animal objects (Figure 2d), in others the animal objects are captured in the whole body but too far from camera (Figure 2b), in varied views or deformations (Figure 2g), or occlusion (Figure 2f). Further more, numerous images are in grayscale as they were captured at night with infrared flash support (Figure 2e), and a large number of images contains no animal as Figure 2h (75% of the Snapshot Serengeti [8] and 32.26% of Wildlife Spotter labeled images were classified as "*no animal*"), while in others might appear several objects belonging to different species. Overwhelming amounts of data and limited image quality, therefore, remarkably slow down the image analyzing process.

To share scientists' workload, in large wildlife monitoring projects such as Snapshot Serengeti or Wildlife Spotter, volunteers were invited as *"citizen scientists"* to join the image analyzing process remotely via Web-based image classification systems. A large number of volunteers engaged in these projects and the species identifying accuracy of 96.6% obtained on the Snapshot Serengeti dataset [8], which was validated by experts, demonstrates the success of citizen science projects.

[1]https://www.snapshotserengeti.org
[2]https://wildlifespotter.net.au



(a) a good image (b) scale variation/far field problem: the *bird* is far from camera

(c) background clutter: the *reptile* blends into scene (d) object is too close to camera

(e) image captured at night with infrared flash (f) object occlusion problem

(g) object deformation/unclear view (h) an image without animal, but being presented to citizen scientist to annotate

Figure 2: Examples from Wildlife Spotter image dataset under various scenarios. Original images are in resolutions of 1920×1080 or 2048×1536 pixels. All images are resized for illustration.

However, the huge collections of images and the limitation of imperfect image quality notably influence human classification speed, and sometimes accuracy, even for experts [8]. In particular, some images in the Snapshot Serengeti dataset were annotated by experts as "*impossible to identify*", over 9,600 images in the Wildlife Spotter dataset of South-central Victoria were tagged as *"something else"* or *"image problem"*, thousands of photos were inconsistently labeled (e.g., the same image was classified as different species by different volunteers). In addition, even though many volunteers were enthusiastic about joining citizen science projects, it would take a long time to complete analyzing millions of images manually. For example, in the Snapshot Serengeti project, it took more than two months to annotate a 6-month batch of images by a group of 28,000 registered and

40,000 unregistered volunteers [8]. The demand of wild animal identifying automation, therefore, arises from these obstacles. To our best of knowledge, there are currently very limited existing works have attempted to build automated system to process and analyze videos and images captured in the wild for environmental monitoring task.

The overwhelming amounts of data from camera traps highlight the need for image processing automation. From data analysis and machine learning point of views, there are some immediate techniques to make wildlife identification automated such as applying linear support vector machine (SVM) classifier with manual object bounding on hand-crafted features [10], convolutional neural network (CNN) model with automatic object detection [11], or fine-tuning CNN models inheriting model weights pretrained on a very large scale dataset such as the ImageNet [12], [13]. These approaches addressed the problem of wildlife monitoring automation and demonstrated promisingly empirical results. However, two primary challenges, which inhibit the feasibility of an automated wildlife monitoring application in practice, are still remaining. The first obstacle is that, to obtain applicable image classification accuracy, an enormous amount of manual pre-processing is still required to input images for detecting and bounding animal objects [10]. The second limitation is poor performance obtained by wildlife monitoring system, in spite of complete automation, requiring much more improvements for practical application [11].

In this paper, we design a framework for animal recognition in the wild, aiming at a fully automatic wildlife spotting system. Our work is motivated by the state-of-the-art power of recent deep CNN models for image classification, in particular the recent evidence that automated recognition can surpass human at certain object recognition tasks in the ImageNet competition [14]. We carry out experiments on datasets of Wildlife Spotter project, containing a large number of images taken by trap cameras set up by Australian scientists. More specifically, since the Wildlife Spotter dataset includes both animal and non-animal images, we divide the wild animal identifying automation into two subsequent tasks: (1) *Wildlife detection,* which is actually a binary classifier capable of classifying input images into two classes: "*animal*" or "*no animal*" based on the prediction of animal presence in images; and (2) *Wildlife identification,* a multiclass classifier to label each input image with animal presence by a specified species. The core of each task is essentially a deep CNN-based classifier, trained from prepared datasets manually labeled by volunteers. Several selected deep CNN architectures are employed to the framework for comparisons. The success of *Task 1* will have a significant impact in improving the efficiency of citizen science-based projects (e.g., Wildlife Spotter) by automatically filtering out a large portion of non-animal images where citizen annotators are currently wasting their time on. Our experimental results on the Wildlife Spotter datasets show that this approach is feasible, and can save considerable time and expense. Hence, the key contribution of this work is that, with sufficient data and computing infrastructure, deep learning could be employed to build a fully automatic image classification system at large scale, liberating scientists from the burden of manual processing of millions of images, which is considered by the project managers "*It's a job that computers just can't do*"[3]. In addition, our proposed framework can be combined with the existing citizen science project, forming a "hybrid" image classifier whose automated component works as a recommendation system, providing volunteers remarkable suggestions to speed up their classifying decisions.

The rest of the paper is organized as follows. In Section II we briefly outline fundamentals of CNN and its application to image classification. In this section we also summarize related work for the topic of automated wildlife classification and an existing citizen science-based wild animal classification project: the Wildlife Spotter. We describe the proposed animal recognition framework, data, and experimental set-up in Section III. Empirical results and discussion are presented in Section IV. Finally we conclude and state future work in Section V.

II. Related work

In this section we first briefly describe the CNN and its application to image classification. We then summarize various CNN architectures that have demonstrated the state-of-the-art performance in recent ImageNet Challenges [14]. Finally we discuss existing approaches to a particular problem: animal classification in natural scenes from camera trap images.

A. Convolutional Neural Networks for Image Classification

Visual recognition is a relatively trivial task for human, but still challenging for automated image recognition systems due to complicated and varied properties of images [15]. Each object of interest can alter an infinite number of different images, generated by variations in position, scale, view, background, or illumination. Challenges become more serious in real-world problems such as wild animal classification from automatic trap cameras, where most captured images are in imperfect quality as described previously in Section I. Therefore, for the task of image classifying automation, it is important to build models that are capable of being invariant to certain transformations of the inputs, while keeping sensitivity with inter-class objects [16].

Firstly proposed by LeCun *et al.* [17], CNNs have been showing great practical performance and been widely used in machine learning in the past recent years, especially in the areas of image classification [14], [18], [19], [20], [21], speech recognition [22], and natural language processing [23], [24]. These models have made the state-of-the-art results that even outperformed human in image recognition task [25], due to recent improvements in neural networks, namely deep CNNs, and computing power, especially the successful implementations of parallel computing on graphical processing units (GPUs), and heterogeneous distributed systems for learning deep models in large scale such as TensorFlow [26].

[3]https://wildlifespotter.net.au/classify

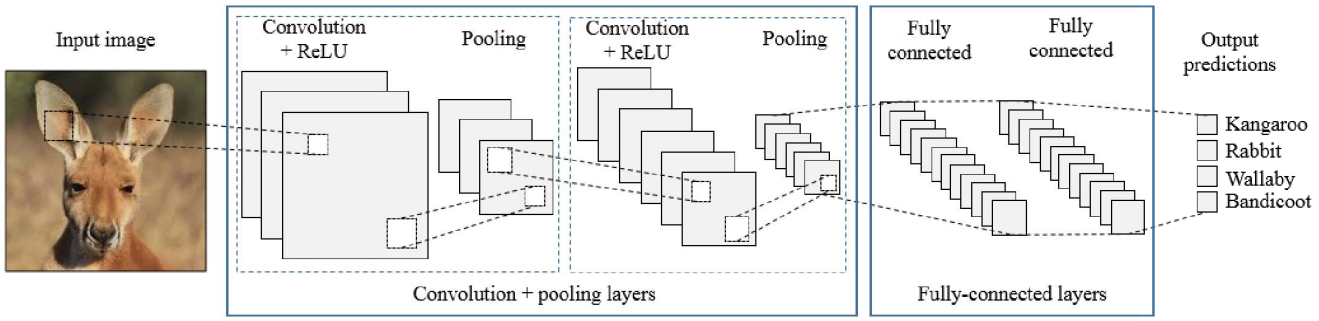


Figure 3: Illustration of a typical convolutional neural network architecture setup.

CNNs are basically neural network-based learning models specifically designed to take advance the *spatial structure* of input images, which are usually in 3-dimensional volume: width, height, and depth (the number of color channels). As illustrated in Figure 3, a CNN is essentially a sequence of layers which can be divided into groups each comprising of convolutional layer plus non-linear activation function, usually the Rectifier Linear Unit (ReLU) [20], and pooling layer, mostly max pooling; ended by several fully-connected layers where the last one is the output layer with predictions. In the standard neural networks, each neuron is fully connected to all neurons in the previous layer and the neurons in each layer are completely independent. When applied to high dimensional data such as natural images, the total number of parameters can reach millions, leading to serious overfitting problem and impractical to be trained. In CNNs, by contrast, each neuron is connected only to a small region of the preceding layer, forming *local connectivity*. The convolution layer computes the outputs of its neurons connected to local regions in the previous layer, the spatial extent of this connection is specified by a filter size. In addition, another important property of CNNs, namely *parameter sharing,* dramatically reduces the number of parameters and so does computing complexity. Thus, compared to regular neural networks with similar size of layers, CNNs have much fewer connections and parameters, making them easier to train while their performance is slightly degraded [20]. These three main characteristics – *spatial structure, local connectivity* and *parameter sharing* – allow CNNs converting input image into layers of abstraction; the lower layers present detail features of images such as edges, curves and corners, while the higher layers exhibit more abstract features of object.

Apart from using more powerful models and better techniques for preventing overfitting, the performance of data-driven machine learning approaches depends strictly on the size and quality of collected training datasets. Real-life objects exhibit considerable variability, requiring much larger training sets to learn recognizing them [20]. The ImageNet, one of the world's largest public image datasets to date, contains over 14 million color, high-resolution, human-labeled images of 22,000 categories. A reduced version of ImageNet dataset including 1,000 categories, each contains roughly 1,000 images, was released in 2010 by the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), has recently been the standard benchmark for evaluation of large scale image classification models [13], [20].

The ILSVRC aims at three main tasks: image classification (from 2010), single-object localization (from 2011), and object detection (from 2013) from the ImageNet dataset [14]. There were many efforts from research groups over the world participating the challenge and the reported performance was improved significantly over time. The winner of ILSVRC-2010 was AlexNet [20], a CNN-based architecture comprising of 8 layers with 5 convolutional layers and 3 fully-connected layers. A variant of the AlexNet model also achieved over 10% top-5 test error rate better than the second-best entry [20]. GoogLeNet [19], the winner of ILSVRC-2014, developed an *Inception Module* that dramatically reduces the number of parameters. Further more, the GoogLeNet replaced fully-connected layers at the top of the CNN by average pooling, removing a large number parameters which do not affect performance of the network. The VGG Nets [18], which are analogous to AlexNet but the network depth was increased up to 19 layers, with smaller convolutional filters, outperformed other models in the ILSVRC-2014 except the GoogLeNet. Not only show great performance on the ImageNet dataset, the VGG models also generalize well and achieve the best results on other datasets [18]. The most recently published state-of-the-art architecture is the ResNet, a residual learning framework with a depth of up to 152 layers but still having lower complexity than the VGG Nets. Similar to the VGG Nets, the ResNet also shows good generalization performance on new datasets other than the ImageNet [21].

B. Wildlife Classification

Monitoring wildlife through camera traps is an effective and reliable method in natural observation as it can collect a large volume of visual data naturally and inexpensively. The wildlife data, which can be fully automatic captured and collected from camera traps, however, is a burden for biologists to analyze to detect whether there exist animal in each image, or identify which species the objects belong to. Making this costly, time-consuming manual analyzing process automated thus could

Table I: The most common and successful CNN architectures for image classification.

Model	Trainable layers	Main specifications
AlexNet	8	5 convolutional layers and 3 fully-connected layers. [20]
VGG-16	16	13 convolutional layers with 3x3 filters, and 3 fully-connected layers. [18]
GoogLeNet	22	Developed an *Inception Module* that dramatically reduces the number of parameters while achieving high accuracy. Average pooling is used at top of CNN instead of fully-connected layers. [19]
ResNet-50	50	A deep residual learning framework, skip connections and batch normalization. Much deeper than VGG-16 (50 compared to 16) but having lower complexity and higher performance. [21]

dramatically reduce a large amount of human resource and quickly provide research findings.

There were few attempts to build an automatic wildlife classification system. In [10], Yu *et al.* employed improved sparse coding spatial pyramid matching (ScSPM) for image classification [27], [28]. Animal objects are first manually detected and croppedout of the background with the whole body, then image features are extracted based on the ScSPM to convert an image or a bounding box to a single vector, finally a linear multi-class SVM is applied for classification. The average classification accuracy was at 82% on their own dataset of 7,196 images of 18 species. Chen *et al.* proposed a CNN-based model with automatic image segmentation preprocessing [11]. The network comprises of three convolutional layers with filter size of 9×9, each followed by a max pooling layer with kernel size of 2×2, ended by a fully-connected layer and a softmax layer. In addition, different to [10], in [11] the animal object cropping process was carried out automatically by applying an automatic segmentation method, namely Ensemble Video Object Cut (EVOC) [29]. Although Chen's proposed framework is fully automatic and outperformed a traditional Bag-of-visual-words model based image classification algorithm [30], [31], the recognition results obtained on their own dataset were only around 38.32%, inapplicable to practice. Motivated by the success of deep CNN-based models in recent ILSVRC contests, Gomez *et al*. [32], [12] employed deep CNN models, which have shown the state-of-the-art performances on the ImageNet dataset, to deal with the problem of large scale wild animal identification on a new open dataset, the Snapshot Serengeti [8]. More specifically, in [32], [12] all CNN models were pre-trained with the ImageNet dataset, then re-trained on top layers of new dataset, namely fine-tuning technique. This comes from the assumption that in data-driven approaches, a network pre-trained on a large dataset such as the ImageNet would have already learned features well for most image classification problems, resulting in better performance than training on smaller datasets [12]. The method achieved their best results at 88.9% and 98.1% for top-1 and top-5 accuracy, respectively [12]. This approach arises a question that, will the CNNs gain better accuracy when training on new large dataset from scratch compared to those inheriting available pre-trained models as [12] did? Additionally, Gomez's approach did not solve the task of automatic animal detection to filter out non-animal images, which should be considered first on datasets containing a large number of images without animal presence.

Inspired by the great success of deep CNN-based models, in this work we apply deep CNN models for wild animal classification, similar with [11] and [12], on the Wildlife Spotter dataset. Different to [12], we solve the task of animal image filtering prior to the task of animal identification, as the Wildlife Spotter dataset contains a large amount of blank images (i.e. images without animal presence). Further more, for the task of animal identification, we investigate two training scenarios for comparisons: training models from scratch on the Wildlife Spotter dataset, and training with available ImageNet pre-trained models (i.e. fine-tuning).

C. Citizen Science

Citizen science plays an important role in many research areas, particularly in ecology and environmental sciences [33], [34], [35], [36]. A citizen scientist is a volunteer who contributes to science by collecting and/or processing data as part of a scientific enquiry. Significant development in digital technique, especially the Internet and mobile computing, is one of key factors responsible for the great explosion of recent citizen science projects [33]. Volunteers are now able to, remotely, take part to a project by using designated applications on their mobile phones or computers to collect data or process introduced data, and then enter them online into centralized, relational databases [36]. Citizen scientists now participate in many projects on a range of areas, including climate change, invasive species and monitoring of all kinds [33], [36]. In addition, the engagement of public significantly facilitates the area of machine learning. Supervised machine learning algorithms require large amounts of labeled data to train automated models, thus human-labeled datasets, such as Snapshot Serengeti or Wildlife Spotter, are valuable resources. Many Internet based applications, such as Google Search, Facebook or Amazon, are leveraging machine learning techniques through data collected from public user activities to enhance their business management.

Apart from valuable contributions the citizen science brings about, several challenges arise when working with citizen science data [36]. Two technical principles, therefore, should be considered. First, data collected by citizen scientists must be properly validated. Second, it is important to design standard methods and tools for data collection and processing [36].

D. Wildlife Spotter Project

Wildlife Spotter is an online citizen science project undertaken by several Australian organizations and universities, taking crowd-sourcing approach to science by asking volunteers to help scientists classifying animals from millions of images

collected from automatic trap cameras. These cameras, located in the nation wide: tropical rainforests, dry rangelands, and around the cities, set up to automatically snap color, high definition images day and night. To date, over 3 million images were completed. To deal with the enormous volume of images, the project invites volunteers playing as "*citizen scientists*" to join image analyzing. The main goal of the project is, through analyzing captured images, to assist researchers study Australian wildlife populations, behaviors and habitats to save threatened species and preserve balanced, diverse, and sustainable ecosystems[4].

The Wildlife Spotter project is divided into six sub-projects, specializing on separated natural areas of Australia: Tasmanian nature reserves, Far north Queensland, South-central Victoria, Northern Territory arid zone, New South Wales coastal forests, and Central mallee lands of New South Wales[5]. Volunteers participate the project by registering online accounts, logging in the Web-based image classification system and manually labeling the displayed images, one by one. User assigns an introduced image to a specific species by clicking the appropriate category from a given list of animals. In case of uncertainty, blank image or image problem, user labels image as *"Something else"*, *"There is no animal in view"* or *"There is a problem with this image"*, respectively. In order to obtain reliable classification accuracy, each image in the dataset is repeatedly introduced to a number of different users to label. For instance, most classified images in the South-central Victoria dataset each was annotated by five citizen scientists. As we described in Section I, the image datasets collected from camera traps are usually in large volume and in imperfect quality, which critically prolong processing time and probably lead to misclassification or inconsistent labeling. In this work, we aim at building a practical, fully automatic animal recognition framework for Wildlife Spotter project, freeing scientists from the burden of manual labeling, while dramatically reducing processing time.

III. Deep CNN for animal recognition framework

In this section, we present our proposed image classification framework and its application to the Wildlife Spotter datasets. First we describe the datasets. Then we introduce a CNN-based framework for wildlife identification. Finally we characterize selected CNN architectures employed in our experiments and implementations.

A. Wildlife Spotter Dataset

We take particular interest in the Wildlife Spotter dataset of South-central Victoria, including 125,621 single-labeled images to date. The images are collected from many different scenes using 30 Reconyx HC600 Hyperfire covert cameras; and are captured at daylight without flash and at night with infrared flash in both color and grayscale settings at 1920×1080 or 2048×1536 resolutions. From this dataset, we take a list of 108,944 labeled images, each annotated by, approximately, 5 different citizen scientists. A citizen scientist was trained to annotate an observed animal as a category among 15 wildlife species in South-central Victoria, Australia (*bandicoot, wombat, rat, brushtail possum, mouse, cat, rabbit, wallaby, ringtail possum, echidna, dog, fox, koala, kangaroo,* and *deer*) and 3 groups of species (*mammal, bird,* and *reptile*). Image without appearance of animal is labeled as *"no animal"*. If the user is not confident with his/her assessment due to bad quality images or occlusions, the image is labeled as *"something else"* or *"image problem",* respectively.

To preprocess the data, we eliminate the samples that are: duplicated or inconsistently labeled (i.e. the same image but being tagged differently by different citizen scientists, e.g., images are labeled as *"something else"* or *"image problem"*, or having the tags *"animal"* and *"no animal"* at the same time). In the end, we obtain a set of 107,022 single-labeled images containing 34,524 non-animal samples and the rest 72,498 samples of 18 species as listed in Table II, this accounts for more than 85% of the whole original dataset.

Next, we construct two settings to apply our proposed framework on two tasks: *Wildlife detection* and *Wildlife identification*. For the former, we consider a binary classification problem and experiment with both balanced and imbalanced classes. Firstly we investigate the typical situation in training machine learning algorithms: the balanced dataset; each training class has equivalently 25,000 samples for training and 8,500 for validation. The balanced dataset preparation is done by under-sampling the superior classes down to the size of the minority class. Data imbalance is a popular phenomenon in real-life problems when some classes are superior to others, and the Wildlife Spotter project is no exception. In particular, the largest population is *bird* with 22,145 samples while the least species, *deer*, appears in only 16 images. This highly imbalance probably leads to misclassification since classifier tends to be biased to superior classes. In case of imbalanced dataset we use 107,000 labeled images for training, divided into two sub-sets: training set and validation set. The training set consists of 80,000 images including 55,000 labeled as *"animal"* and the remaining 25,000 labeled as *"no animal"*; the validation set includes 18,500 and 8,500 images labeled as *"animal"* and *"no animal"*, respectively.

For the later case of *Wildlife identification* or animal recognition, due to a large number of animals with different number of observations (cf. Table II), we resort to two experimental cases: identifying the three most common species and the six most common species respectively. In case of identifying the three most common species (*bird, bandicoot,* and *rat*), we first investigate the case of balanced dataset, where each class consists of 8,000 images for training and 2,000 for validation. We then carry out training and testing on all samples of these three classes from the dataset, i.e. the imbalanced dataset case. The list of species and corresponding number of images used for the imbalanced case is listed in Table II. For the most complicated task: identifying the six most common species (*bird, rat, bandicoot, rabbit, wallaby,* and *mammal*), we investigate only the case of imbalanced dataset; all samples

[4]https://wildlifespotter.net.au/about

[5]https://wildlifespotter.net.au/projects

Table II: Representative of animals with label data from Wildlife Spotter dataset of South-central Victoria, Australia (sorted in descending order of the number of images).

Species	Samples	Species	Samples
Bird	22145	*Brushtail Possum*	1072
Rat	11884	*Wombat*	616
Bandicoot	10507	*Kangaroo*	322
Rabbit	7965	*Echidna*	307
Wallaby	5370	*Ringtail Possum*	274
Mammal	4982	*Dog*	204
Mouse	3901	*Reptile*	158
Cat	1531	*Koala*	27
Fox	1217	*Deer*	16

from the six species will be used for training, 80% for training and the rest 20% for validation.

B. Recognition Framework for Animal Monitoring

As described above in this section, the Wildlife Spotter labeled dataset contains both animal and non-animal images with proportions of 67.74% and 32.26%, respectively. This fact arises two tasks of Wildlife Spotting system: (1) *Wildlife detection* to specify whether there exist animal in an image, and (2) *Wildlife identification* to identify which species the animal objects belong to. It has been shown that CNNs outperform other approaches in the topic of image classification; thus in this work we focus on adopting recent state-of-the-art CNN architectures for both those two tasks – detection and recognition.

As depicted in Figure 4, our proposed recognition system consists of two CNN-based image classification models corresponding to the two addressed tasks. First a CNN-based model is designed to train a binary classifier, namely *Wildlife detector*; then another CNN-based model is created to train a multi-class classifier, namely *Wildlife identifier*.

1) CNN Architectures: Three CNN architectures with different depths are employed to our proposed framework, namely Lite AlexNet, VGG-16 [18], and ResNet-50 [21]. We use a simplified version of AlexNet [20] and call it Lite AlexNet, with less hidden layers and feature maps at each layer. In particular, the Lite AlexNet comprises of three 2-D convolutional layers with ReLU activations and MaxPooling, followed by two fully-connected layers: one with ReLU nonlinear activation plus Dropout for reducing overfitting, the output layer with sigmoid activation for binary classification in detecting task and softmax activation for multiclass classification in recognizing task. All convolutional layers have small filter size of 3×3, while all max-pooling layers have window size of 2×2 pixels. VGG-16 and ResNet-50 are two representatives of the state-of-the-art CNN architectures that not only showed excellent performances on the ILSVRC [20], but also generalized well to other datasets [18], [21]. The input to all CNN architectures is a fixed-size 224×224 image in RGB color.

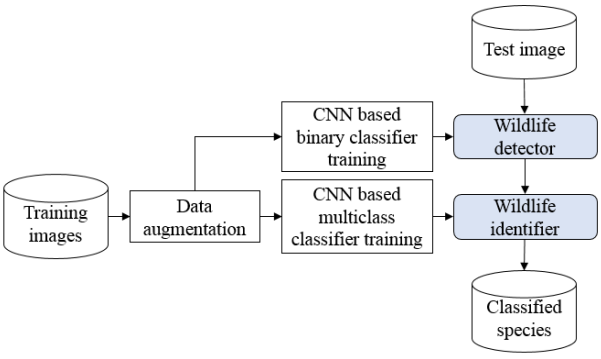


Figure 4: Key steps in the proposed framework for automated wild animal identification.

2) Image Processing: The Wildlife Spotter dataset contains high resolution images of 1920×1080 and 2048×1536 pixels, while the input of CNN models must be in fixed dimension. Therefore, in our experiments all original images were downscaled to 224×224 pixels for training. In [20] this process was carried out by firstly rescaling the shorter side of image to the fixed length, then applying center cropping the image with the same length. In this work, for simplicity, we rescale both image width and height simultaneously, which may result in image distortion. Pixel intensities are normalized into the range of $[0, 1]$. Data quality, which can be enhanced by augmentation techniques, is a key to data-driven machine learning models; however in this work a few data augmentation processes, shearing and zooming, were applied to training images.

3) Training Deep Networks: Our implementation is in Keras [37], a high-level neural networks API, with TensorFlow backend [26]. Adam optimizer, the first-order gradient-based optimization based on adaptive estimates of lower-order moments, was employed for training all networks [38]. A small minibatch size of 16 was set to all experiments. We train our models on four NVIDIA Titan X GPUs, each network takes three to five days to finish training.

For each task we train CNN models in two scenarios: imbalanced and balanced datasets. We compute classification accuracy for both cases. In case of dataset imbalance, F-measure is employed in addition to accuracy, to test the robustness of the proposed system. Accuracy on the validation set is used as performance metric. To evaluate transfer learning, we carry out training *Task 2 – Wildlife identification*, in two scenarios: training model from scratch and fine-tuning with available ImageNet pre-trained models. Fine-tuning techniques leverage a network pre-trained on a large dataset, in this case is the ImageNet, based on the assumption that such network would have already learned useful features for most computer vision problems, thus could reach better accuracy than a model trained on a smaller dataset. Our fine-tuning process follows three steps: firstly the convolutional blocks are instantiated, then the model will be trained once on new training and validation data, finally the fully-connected model with fewer specified classes will be trained on top of the stored features [37].

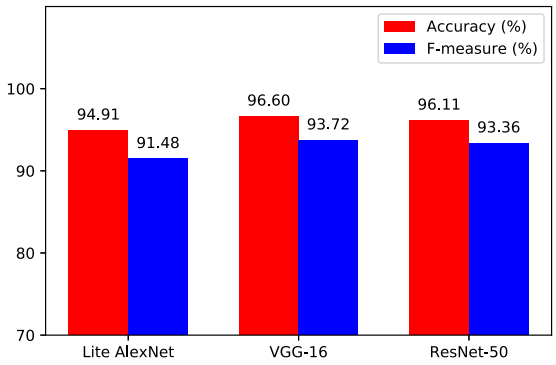


Figure 5: Animal vs. Non-animal image detection accuracy on Wildlife Spotter dataset of South-central Victoria, Australia. The data are imbalanced; the training set contains 55,000 animal images and 25,000 non-animal images, the validation set contains 18,500 animal images and 8,500 non-animal images. F-measure was used, in addition to accuracy, to evaluate the system's robustness.

IV. Experimental results and discussion

A. Results for Recognizing Animal vs. Non-animal Images

Figure 5 shows the performance of *Task 1* with three different CNN architectures on Wildlife Spotter imbalanced dataset. Overall, all models achieved very high results. In particular, the best accuracy stands at 96.6% (with VGG-16 architecture), ResNet-50 comes next at 95.96%, which is only marginally lower. These results, once again, confirm that VGG and ResNet models generalize well to other datasets as observed in [18], [21]. In addition, Lite AlexNet, the simplest architecture with only five learnable layers, also showed excellent result with this binary classification task. Similarly, high performance achieved with F-measure metrics indicates that these models are robust against imbalanced data.

To further test whether imblanced data is a notable issue for our animal recognition problem, Table III shows the performance on balanced dataset described earlier. We keep three CNN architectures the same as in the case of data imbalance. As can be observed, for all models, the performances were only marginally degraded, which might due to under-sampling process, where the samples of superior classes were considerably reduced to obtain a balanced dataset. This is again, confirming the promise of the method where it is robust with very high accuracy in detecting images with animals.

Table III: Animal vs. Non-animal image detection accuracy on Wildlife Spotter dataset of South-central Victoria, Australia. The data are balanced, each class contains 25,000 images for training and 8,500 for validation.

Model	Accuracy (%)
Lite AlexNet	92.68
VGG-16	95.88
ResNet-50	95.65

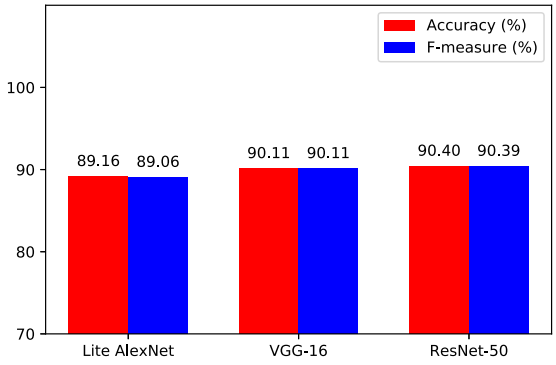


Figure 6: Animal identification accuracy on Wildlife Spotter dataset of the three most common species (*bird*, *rat,* and *bandicoot*). The training set is imbalanced as listed in Table II, 80% images of each class are used for training, 20% for validation.

B. Animal Identification Results

1) Identifying the three most common species: For training the model for identifying the three most common species (*bird*, *rat,* and *bandicoot*) in case of imbalanced dataset, all samples from these species are used for training. The training set contains 80% images of each class (35,629 images), the validation set is the rest 20% (8,907 images). As shown in Figure 6, the animal identifying task achieved very high performance for all CNN architectures with accuracy ranging from 89.16% to 90.4%. The simplest model, Lite AlexNet, demonstrates the lowest performance. The deepest model, ResNet-50, shows the best results, however the first runner-up, VGG-16, shows very close performance.

The experimental results of identifying the three most common species, in case of balanced dataset, are showed in Table IV. Having trained from scratch, all three CNN models show comparable performances with classification accuracy ranging from 87.80% to 88.03%. In this case, VGG-16 outperformed the others although the gap is very small. In comparison to performance of *Task 1*, the *Task 2* demonstrates worse results for all models. The possible reason for performance deterioration is two folds, more complicated problem caused by an increased number of classes, while a smaller number of training samples generated by under-sampling process, making the models more difficult to fit the datasets.

Fine-tuning technique was applied only to VGG-16 and ResNet-50 since these models have pre-trained weights available on the ImageNet, and our experimental results show opposite trends. While VGG-16 model achieved a little accuracy improvement of 0.2% on new dataset compared to training from the scratch, the ResNet-50 shows a serious performance drop, from 87.97% down to 76.43% in accuracy, indicating that overfitting might occurred. The most important contribution the fine-tuning technique brought to the framework is the cost of computing. In particular, to run each VGG-16 model

Table IV: Animal identification accuracy on 3 most common species of Wildlife Spotter dataset (*bird*, *rat,* and *bandicoot*) with different CNN architectures. The dataset is balanced. The models are trained in two scenarios: (1) training from scratch, and (2) training with ImageNet pre-trained weights inheritance (i.e. fine-tuning). Note that fine-tuning was not applied to Lite AlexNet due to the unavailability of pre-trained network.

Model	**Accuracy (%)**	
	Training from scratch	***Fine-tuning***
Lite AlexNet	87.80	-
VGG-16	88.03	88.23
ResNet-50	87.97	76.43

training epoch on our learning system, training from scratch took more than 4,000 seconds in general to complete, while fine-tuning did only in around 65 seconds, 61 times faster.

2) Identifying the six most common species: Towards a fully automated wild animal recognition system, we investigate further with the case of identifying the six most common classes from the Wildlife Spotter dataset (*bird*, *rat*, *bandicoot*, *rabbit*, *wallaby*, and *mammal*). In this case we only consider the case of imbalanced dataset, all samples from these six classes are used as listed in Table II, each class is split into two sets, 80% images for training and 20% for validation. Figure 7 shows the wildlife identification results obtained by the three CNN architectures. Again, the system achieves reasonably good results for up to six different types of animal classes. In this case the ResNet-50 appears to be the best model with accuracy at 84.39%, despite the small gap, indicating that deeper CNN architectures would produce better performance in complex recognition problems. In addition, similar performances achieved with F-measure metrics indicates the robustness of the proposed system against imbalanced data.

C. Discussion

The experimental results have shown that high accuracy can be achieved in detecting images that contain animals with more than 96% accuracy. For citizen science-based projects and systems, human annotators' time are valuable and this framework has the promise to radically improve the efficiency in these systems. For example, for the current Wildlife Spotter system up to the time we collected this dataset, more than 32% of images presented to human annotators does not contain any animal to be annotated.

The animal identification results have shown a good performance in identifying the most common three and six species of animals. While this performance may not yet be sufficient to build a fully automatic recognition, it still adds an enormous value in improving the system by automatically providing initial animal labels for human annotators. We anticipate that with more data collected over time and with a rapid growing capacity of deep learning techniques in computer vision, this performance could be improved significantly in near future.

In addition, we also believe that there are different ways that we could use the current dataset from the Wildlife Spotter

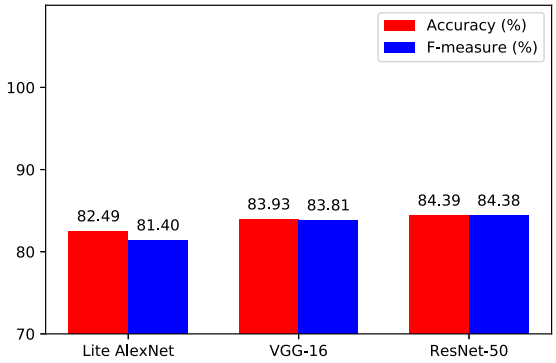


Figure 7: Animal identification accuracy on Wildlife Spotter dataset of the six most common species. The dataset is imbalanced as listed in Table II. From each class 80% images are used for training and the 20% for validation.

system to improve the performance. For example, the training dataset could further be prepared and confirmed by ecology experts, similar to the Golden Standard set in the Snapshot Serengeti project, to enhance the quality of the training data. Besides, data enhancement techniques can be applied to obtain better results; in this work only few data augmentation processes were applied as described in Section III. These will be part of our future work and report.

V. Conclusion

Efficient and reliable monitoring of wild animals in their natural habitats is essential to inform conservation and management decisions. In this paper, using the Wildlife Spotter dataset, which contains a large number of images taken by trap cameras in South-central Victoria, Australia, we proposed and demonstrated the feasibility of a deep learning approach towards constructing scalable automated wildlife monitoring system. Our models achieved more than 96% in recognizing images with animals and close to 90% in identifying three most common animals (*bird*, *rat* and *bandicoot*). Furthermore, with different experimental settings for balanced and imbalanced, the system has shown to be robust, stable and suitable for dealing with images captured from the wild.

We are working on alternative ways to improve the system's performance by enhancing the dataset, applying deeper CNN models and exploiting specific properties of camera trap images. Towards a fully automated wild animal recognition system, we would investigate transfer learning to deal with problem of highly imbalanced data. In the near future, we focus on developing a "hybrid" wild animal classification framework whose automated module working as a recommendation system for the existing citizen science-based Wildlife Spotter project.

Acknowledgment

This work is partially supported by the Telstra-Deakin Centre of Excellence in Big Data and Machine Learning.

REFERENCES

[1] P. M. Vitousek, H. A. Mooney, J. Lubchenco, and J. M. Melillo, "Human domination of Earth's ecosystems," *Science*, vol. 277, no. 5325, pp. 494–499, 1997.

[2] G. C. White and R. A. Garrott, *Analysis of wildlife radio-tracking data*. Elsevier, 2012.

[3] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, 2004, pp. 214–226.

[4] B. J. Godley, J. Blumenthal, A. Broderick, M. Coyne, M. Godfrey, L. Hawkes, and M. Witt, "Satellite tracking of sea turtles: Where have we been and where do we go next?" *Endangered Species Research*, vol. 4, no. 1-2, pp. 3–22, 2008.

[5] I. A. Hulbert and J. French, "The accuracy of GPS for wildlife telemetry and habitat mapping," *Journal of Applied Ecology*, vol. 38, no. 4, pp. 869–878, 2001.

[6] R. Kays, S. Tilak, B. Kranstauber, P. A. Jansen, C. Carbone, M. J. Rowcliffe, T. Fountain, J. Eggert, and Z. He, "Monitoring wild animal communities with arrays of motion sensitive camera traps," *arXiv:1009.5718*, 2010.

[7] A. F. O'Connell, J. D. Nichols, and K. U. Karanth, *Camera traps in animal ecology: Methods and Analyses*. Springer Science & Business Media, 2010.

[8] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer, "Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna," *Scientific Data*, vol. 2, p. 150026, 2015.

[9] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," *Nature*, vol. 381, no. 6582, p. 520, 1996.

[10] X. Yu, J. Wang, R. Kays, P. A. Jansen, T. Wang, and T. Huang, "Automated identification of animal species in camera trap images," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, pp. 1–10, 2013.

[11] G. Chen, T. X. Han, Z. He, R. Kays, and T. Forrester, "Deep convolutional neural network based species recognition for wild animal monitoring," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 858–862.

[12] A. Gómez, A. Salazar, and F. Vargas, "Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks," *arXiv:1603.06169*, 2016.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[14] O. Russakovsky, J. Deng, H. Su *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[15] N. Pinto, D. D. Cox, and J. J. DiCarlo, "Why is real-world visual object recognition hard?" *PLOS Computational Biology*, vol. 4, no. 1, p. e27, 2008.

[16] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, pp. 1–58, 2006.

[17] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[22] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008, pp. 160–167.

[23] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A Convolutional Encoder Model for Neural Machine Translation," *arXiv:1611.02344*, 2016.

[24] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional Sequence to Sequence Learning," *ArXiv e-prints*, 2017.

[25] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3642–3649.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *arXiv:1603.04467*, 2016.

[27] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1794–1801.

[28] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3360–3367.

[29] X. Ren, T. X. Han, and Z. He, "Ensemble video object cut in highly dynamic scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1947–1954.

[30] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[31] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 524–531.

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

[33] J. Silvertown, "A new dawn for citizen science," *Trends in Ecology & Evolution*, vol. 24, no. 9, pp. 467–471, 2009.

[34] R. Bonney, C. B. Cooper, J. Dickinson, S. Kelling, T. Phillips, K. V. Rosenberg, and J. Shirk, "Citizen science: A developing tool for expanding science knowledge and scientific literacy," *BioScience*, vol. 59, no. 11, pp. 977–984, 2009.

[35] A. Irwin, *Citizen science: A study of people, expertise and sustainable development*. Psychology Press, 1995.

[36] J. L. Dickinson, B. Zuckerberg, and D. N. Bonter, "Citizen science as an ecological research tool: Challenges and benefits," *Annual Review of Ecology, Evolution, and Systematics*, vol. 41, pp. 149–172, 2010.

[37] F. Chollet, "Keras," https://github. com/fchollet/keras, 2015.

[38] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.