Check for updates

Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning

Mohammad Sadegh Norouzzadeh[a], Anh Nguyen[b], Margaret Kosmala[c], Alexandra Swanson[d], Meredith S. Palmer[e], Craig Packer[e], and Jeff Clune[a,f,1]

[a]Department of Computer Science, University of Wyoming, Laramie, WY 82071; [b]Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849; [c]Department of Organismic and Evolutionary Biology, Harvard University, Cambridge, MA 02138; [d]Department of Physics, University of Oxford, Oxford OX1 3RH, United Kingdom; [e]Department of Ecology, Evolution, and Behavior, University of Minnesota, St. Paul, MN 55108; and [f]Uber AI Labs, San Francisco, CA 94103

Edited by James A. Estes, University of California, Santa Cruz, CA, and approved April 30, 2018 (received for review November 7, 2017)

Having accurate, detailed, and up-to-date information about the location and behavior of animals in the wild would improve our ability to study and conserve ecosystems. We investigate the ability to automatically, accurately, and inexpensively collect such data, which could help catalyze the transformation of many fields of ecology, wildlife biology, zoology, conservation biology, and animal behavior into "big data" sciences. Motion-sensor "camera traps" enable collecting wildlife pictures inexpensively, unobtrusively, and frequently. However, extracting information from these pictures remains an expensive, time-consuming, manual task. We demonstrate that such information can be automatically extracted by deep learning, a cutting-edge type of artificial intelligence. We train deep convolutional neural networks to identify, count, and describe the behaviors of 48 species in the 3.2 million-image Snapshot Serengeti dataset. Our deep neural networks automatically identify animals with >93.8% accuracy, and we expect that number to improve rapidly in years to come. More importantly, if our system classifies only images it is confident about, our system can automate animal identification for 99.3% of the data while still performing at the same 96.6% accuracy as that of crowdsourced teams of human volunteers, saving >8.4 y (i.e., >17,000 h at 40 h/wk) of human labeling effort on this 3.2 million-image dataset. Those efficiency gains highlight the importance of using deep neural networks to automate data extraction from camera-trap images, reducing a roadblock for this widely used technology. Our results suggest that deep learning could enable the inexpensive, unobtrusive, high-volume, and even real-time collection of a wealth of information about vast numbers of animals in the wild.

deep learning | deep neural networks | artificial intelligence | camera-trap images | wildlife ecology

To better understand the complexities of natural ecosystems and better manage and protect them, it would be helpful to have detailed, large-scale knowledge about the number, location, and behaviors of animals in natural ecosystems (2). Placing motion-sensor cameras called "camera traps" in natural habitats has transformed wildlife ecology and conservation over the last two decades (3). These camera traps have become an essential tool for ecologists, enabling them to study population sizes and distributions (4) and evaluate habitat use (5). While they can take millions of images (6–8), extracting knowledge from these camera-trap images is traditionally done by humans (i.e., experts or a community of volunteers) and is so time-consuming and costly that much of the valuable knowledge in these big data repositories remains untapped. For example, currently it takes 2–3 mo for thousands of "citizen scientists" (1) to label each 6-mo batch of images for the Snapshot Serengeti (SS) dataset. By 2011, there were at least 125 camera-trap projects worldwide (6), and, as digital cameras become better and cheaper, more projects will put camera traps into action. Most of these projects, however, are not able to recruit and harness a huge volunteer force as SS has done to extract information of interest. Even if they are able to extract the information they originally intended to capture, there may be other important data that could be extracted for other studies that were not originally envisioned (e.g., information on nonfocal animal species). Automating the information extraction procedure (Fig. 1) will thus make vast amounts of valuable information more easily available for ecologists to help them perform their scientific, management, and protection missions.

In this work, we focus on harnessing computer vision to automatically extract the species, number, presence of young, and behavior (e.g., moving, resting, or eating) of animals, which are statistics that wildlife ecologists have previously decided are informative for ecological studies based on SS data (9–12). These tasks can be challenging even for humans. Images taken from camera traps are rarely perfect, and many images contain animals that are far away, too close, or only partially visible (Fig. 2 *A*–*C*). In addition, different lighting conditions, shadows, and weather can make the information-extraction task even harder (Fig. 2*D*). Human-volunteer species and count labels are estimated to be 96.6% and 90.0% accurate, respectively, vs. labels provided by experts (1).

Significance

Motion-sensor cameras in natural habitats offer the opportunity to inexpensively and unobtrusively gather vast amounts of data on animals in the wild. A key obstacle to harnessing their potential is the great cost of having humans analyze each image. Here, we demonstrate that a cutting-edge type of artificial intelligence called deep neural networks can automatically extract such invaluable information. For example, we show deep learning can automate animal identification for 99.3% of the 3.2 million-image Snapshot Serengeti dataset while performing at the same 96.6% accuracy of crowdsourced teams of human volunteers. Automatically, accurately, and inexpensively collecting such data could help catalyze the transformation of many fields of ecology, wildlife biology, zoology, conservation biology, and animal behavior into "big data" sciences.

Author contributions: M.S.N., A.N., and J.C. designed research; M.S.N. performed research; M.S.N. contributed analytic tools; M.S.N., A.N., M.K., A.S., M.S.P., C.P., and J.C. analyzed data; and M.S.N., A.N., M.K., A.S., M.S.P., C.P., and J.C. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

This open access article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

Data deposition: Both the code and the models used in this study can be accessed on GitHub at https://github.com/Evolving-AI-Lab/deep_learning_for_camera_trap_images.

[1]To whom correspondence should be addressed. Email: jeffclune@uwyo.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1719367115/-/DCSupplemental.

Published online June 5, 2018.

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.

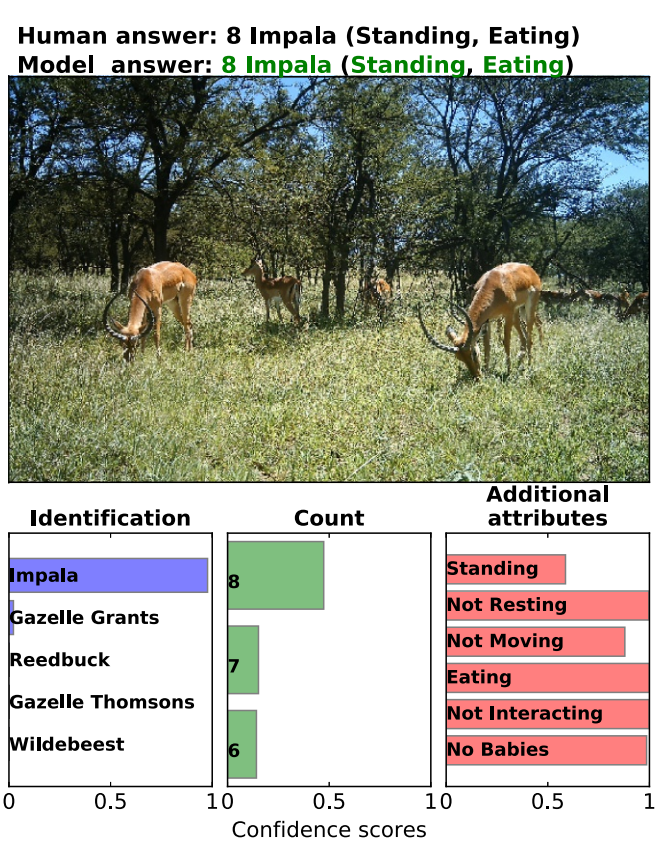

Fig. 1. Deep neural networks (DNNs) can successfully identify, count, and describe animals in camera-trap images. Above the image: The ground-truth, human-provided answer (top line) and the prediction (second line) by a DNN we trained (ResNet-152). The three plots below the image, from left to right, show the neural network's prediction for the species, number, and behavior of the animals in the image. The horizontal color bars indicate how confident the neural network is about its predictions. All similar images in this work are from the SS dataset (1).

Automatic animal identification and counting could improve all biology missions that require identifying species and counting individuals, including animal monitoring and management, examining biodiversity, and population estimation (3). In this work, we harness deep learning, a state-of-the-art machine-learning technology that has led to dramatic improvements in artificial intelligence (AI) in recent years, especially in computer vision (13). Here, we do not harness the data we automatically extract to test a specific ecological hypothesis. Instead, we investigate the efficacy of deep learning to enable many future such studies by offering a far less expensive way to provide the data from large-scale camera-trap projects that has previously led to many informative ecological studies (9–12).

Deep learning only works well with lots of labeled data, significant computational resources, and modern neural network architectures. Here, we combine the millions of labeled data from the SS project, modern supercomputing, and state-of-the-art deep neural network (DNN) architectures to test how well deep learning can automate information extraction from camera-trap images. We find that the system is both able to perform as well as teams of human volunteers on a large fraction of the data and identifies the few images that require human evaluation. The net result is a system that dramatically improves our ability to automatically extract valuable knowledge from camera-trap images. Like every method, deep learning has biases (discussed below) that must be kept in mind, corrected, and/or accounted for when using this technology. Swanson et al., 2016 (14) showed that the citizen-scientist approach also has its own set of systematic biases, but that they can be adequately corrected for.

Background and Related Work

Machine Learning. Machine learning enables computers to solve tasks without being explicitly programed to solve them (15). State-of-the-art methods teach machines via supervised learning (i.e., by showing them correct pairs of inputs and outputs) (16). For example, when classifying images, the machine is trained with many pairs of images and their corresponding labels, where the image is the input and its correct label (e.g., "buffalo") is the output (Fig. 3).

Deep Learning. Deep learning (17) allows computers to automatically extract multiple levels of abstraction from raw data (Fig. 3). Inspired by the mammalian visual cortex (18), deep convolutional neural networks (deep CNNs) are a class of feedforward DNNs (17) in which each layer of neurons (to be "deep," three or more layers) uses convolutional operations to extract information from overlapping small regions coming from the previous layers (13). For classification, the final layer of a DNN is usually a softmax function, with an output between 0 and 1 per class and with all of the class outputs summing to 1. These outputs are often interpreted as the DNN's estimated probability of the image belonging in a certain class, and higher probabilities are often interpreted as the DNN being more confident that the image is of that class (19). DNNs have dramatically improved the state of the art in many challenging problems (13), including speech recognition (20–22), machine translation (23, 24), image recognition (25, 26), and playing Atari games (27).

Related Work. There have been many attempts to automatically identify animals in camera-trap images; however, many relied on

Fig. 2. Various factors make identifying animals in the wild hard even for humans (trained volunteers achieve 96.6% accuracy vs. experts).

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.

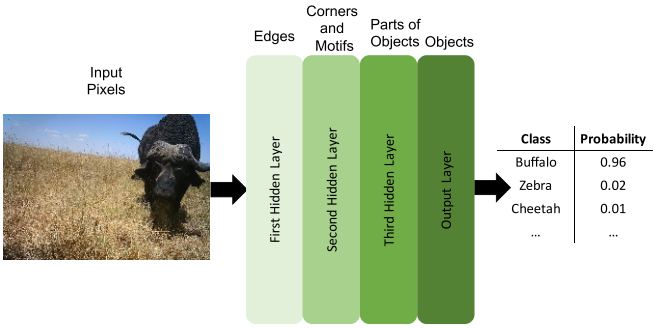

Fig. 3. DNNs have several layers of abstraction that tend to gradually convert raw data into more abstract concepts. For example, raw pixels at the input layer are first processed to detect edges (first hidden layer), then corners and textures (second hidden layer), then object parts (third hidden layer), and so on if there are more layers, until a final prediction is made by the output layer. Note that which types of features are learned at each layer are not human-specified, but emerge automatically as the network learns how to solve a given task.

hand-designed features (8, 28) to detect animals, or were applied to small datasets (e.g., only a few thousand images) (29, 30). In contrast, in this work, we seek to (*i*) harness deep learning to automatically extract necessary features to detect, count, and describe animals; and (*ii*) apply our method on the world's largest dataset of wild animals (i.e., the SS dataset) (1). Reasons to learn features from raw data include that doing so often substantially improves performance (13, 25, 31); because such features can be transferred to other domains with small datasets (32, 33); because it is time-consuming to manually design features; and because a general algorithm that learns features automatically can improve performance on very different types of data [e.g., sound (20, 34) and text (23, 35)], increasing the impact of the approach. However, an additional benefit to deep learning is that if hand-designed features are thought to be useful, they can be included as well in case they improve performance (36–40).

Previous efforts to harness hand-designed features to classify animals include Swinnen et al. (8), who attempted to distinguish the camera-trap recordings that do not contain animals or the target species of interest by detecting the low-level pixel changes between frames. Yu et al. (29) extracted the features with sparse coding spatial pyramid matching (41) and used a linear support vector machine (16) to classify the images. While achieving 82% accuracy, their technique requires manual cropping of the images, which requires substantial human effort.

Several recent works harnessed deep learning to classify camera-trap images. Chen et al. (30) harnessed CNNs to fully automate animal identification. However, they demonstrated the techniques on a dataset of ∼20,000 images and 20 classes, which is of much smaller scale than we explore here (30). In addition, they obtained an accuracy of only 38%, which leaves much room for improvement. Interestingly, Chen et al. found that DNNs outperform a traditional Bag of Words technique (42, 43) if provided sufficient training data (30). Similarly, Gomez et al. (44) also had success applying DNNs to distinguishing birds vs. mammals in a small dataset of 1,572 images and distinguishing two mammal sets in a dataset of 2,597 images.

The closest work to ours is Gomez et al. (45), who also evaluate DNNs on the SS dataset: They perform only the species-identification task, whereas we also attempt to count animals, describe their behavior, and identify the presence of young. On the species-identification task, our models performed far superiorly to theirs: 92.0% for our best network vs. ∼57% (estimating from their plot, as the exact accuracy was not reported) for their best network. There are multiple other differences between our work and theirs. (*i*) Gomez et al. (45) only trained networks on a simplified version of the full 48-class SS dataset. Specifically, they removed the 22 classes that have the fewest images (*SI Appendix*, Fig. S8, bottom 22 classes) from the full dataset and thus classified only 26 classes of animals. Here, we instead sought solutions that performed well on all 48 classes, as the ultimate goal of our research is to automate as much of the labeling effort as possible. (*ii*) Gomez et al. (45) based their classification solutions on networks pretrained on the ImageNet dataset (46), a technique known as transfer learning (32). We found that transfer learning made very little difference on this task when training on the full dataset (*SI Appendix*, *Transfer Learning*), and we thus chose not to use it for simplicity. We revisit the benefits of transfer learning on smaller datasets below. We conduct a more detailed comparison with Gomez et al. (45) in *SI Appendix*, *Comparing to Gomez et al., 2016*.

SS Project. The SS project is the world's largest camera-trap project published to date, with 225 camera traps running continuously in Serengeti National Park, Tanzania, since 2011 (1). Whenever a camera trap is triggered, such as by the movement of a nearby animal, the camera takes a set of pictures (usually three). Each trigger is referred to as a capture event. The public dataset used in this work contains 1.2 million capture events (3.2 million images) of 48 different species.

Nearly 28,000 registered and 40,000 unregistered volunteer citizen-scientists have labeled 1.2 million SS capture events. For each image set, multiple users label the species, number of individuals, various behaviors (i.e., standing, resting, moving, eating, or interacting), and the presence of young. In total, 10.8 million classifications from volunteers have been recorded for the entire dataset. Swanson et al. (1) developed a simple algorithm to aggregate these individual classifications into a final "consensus" set of labels, yielding a single classification for each image and a measure of agreement among individual answers. In this work, we focus on capture events that contain only one species; we thus removed events containing more than one species from the dataset (1.2% of the events). Extending these techniques to images with multiple species is a fruitful area for future research. In addition to volunteer labels, for 3,800 capture events, the SS dataset also contains expert-provided labels, but only of the number and type of species present.

We found that 75% of the capture events were classified as empty of animals. Moreover, the dataset is very unbalanced, meaning that some species are much more frequent than others (*SI Appendix*, *Improving Accuracy for Rare Classes*). Such imbalance is problematic for machine-learning techniques because they become heavily biased toward classes with more examples. If the model just predicts the frequent classes such as wildebeest or zebra most of the time, it can still get a very high accuracy without investing in learning rare classes, even though these can be of more scientific interest. The imbalance problem also exists for describing behavior and identifying the presence of young. Only 1.8% of the capture events are labeled as containing babies, and only 0.5% and 8.5% of capture events are labeled as interacting and resting, respectively. We delve deeper into this problem in *SI Appendix*, *Improving Accuracy for Rare Classes*.

The volunteers labeled entire capture events (not individual images). While we do report results for labeling entire capture events (*SI Appendix*, *Classifying Capture Events*), in our main experiment, we focused on labeling individual images instead because if we ultimately can correctly label individual images, it is easy to infer the labels for capture events. Importantly, we also found that using individual images resulted in higher accuracy because it allowed three times more labeled training examples (*SI Appendix*, *Classifying Capture Events*). In addition,

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.

training our system on images makes it more informative and useful for other projects, some of which are image-based and not capture-event-based.

However, the fact that we took the labels for each capture event and assigned them to all of the individual images in that event introduced noise into the training process. For example, a capture event may have had one image with animals, but the remaining images empty (Fig. 4). Assigning a species label (e.g., hartebeest; Fig. 4*A*) to all these images (Fig. 4 *B* and *C*) added some noise that machine learning models had to overcome.

Experiments and Results

We found that a two-stage pipeline outperformed a one-step pipeline (*SI Appendix*, *One-Stage Identification*): In the first stage, a network solved the empty vs. animal task (task I) (i.e., detecting if an image contains an animal); in the second information-extraction stage, a network then reported information about the images that contain animals. We found that 75% of the images were labeled empty by humans; therefore, automating the first stage alone saves 75% of human labor.

The information-extraction stage contains three additional tasks: task II, identifying which species is present; task III, counting the number of animals; and task IV, describing additional animal attributes (their behavior and whether young are present). We chose to train one model to simultaneously perform all of these tasks—a technique called multitask learning (47)—because (*i*) these tasks are related, therefore they can share weights that encode features common to all tasks (e.g., features that help recognize animals); learning multiple, related tasks in parallel often improves the performance on each individual task (48); and (*ii*) doing so requires fewer model parameters vs. a separate model for each task, meaning we can solve all tasks faster and more energy-efficiently, and the model is easier to transmit and store. These advantages will become especially important if such neural network models run on remote camera traps to determine which pictures to store or transmit.

Datasets. In this work, we only tackled identifying one instead of multiple species in an image [i.e., single-label classification (16)]. Therefore, we removed images that humans labeled as containing more than one species from our training and testing sets (1.2% of the dataset). The training and test sets for the information extraction stage were formed from the 25% of images that were labeled as nonempty by humans.

If there are overly similar images in the training and test sets, models can just memorize the examples and then do not generalize well to dissimilar images. To avoid this problem, we put entire capture events (which contain similar images) into either the training or test set. From a total of 301,400 capture events that contained an animal, we created a training set containing 284,000 capture events and two test sets. The expert-labeled test set contains 3,800 capture events with species and counts labels. The volunteer-labeled test set contains 17,400 capture events labeled by volunteers, and it has labels for species, counts, behaviors, and the presence of young. The dataset contains images taken at day and at night, but we found this had little effect on performance (*SI Appendix*, *Day vs. Night Accuracy*).

Architectures. Different DNNs have different architectures, meaning the type of layers they contain (e.g., convolutional layers, fully connected layers, pooling layers, etc.) and the number, order, and size of those layers (13). In this work, we tested nine different modern architectures at or near the state of the art (Table 1) to find the highest-performing networks and to compare our results to those from Gomez et al. (45). We only trained each model one time because doing so is computationally expensive and because both theoretical and empirical evidence suggests that different DNNs trained with the same architecture, but initialized differently, often converge to similar performance levels (13, 17, 51).

A well-known method for further improving classification accuracy is to use an ensemble of models at the same time and average their predictions. After training all of the nine models for each stage, we formed an ensemble of the trained models by averaging their predictions (*SI Appendix*, *Prediction Averaging*). More details about the architectures, training methods, preprocessing steps, and the hyperparameters are in *SI Appendix*, *Preprocessing and Training*. To enable other groups to replicate our findings and harness this technology for their own projects, we are publishing the software required to run our experiments as freely available, open-source code. We are also publishing the final DNNs trained on SS so that others can use them as is or for transfer learning. Both the code and the models can be accessed at https://github.com/Evolving-AI-Lab/deep_learning_for_camera_trap_images.

Task I: Detecting Images That Contain Animals. For this task, our models took an image as input and output two probabilities describing whether the image had an animal or not (i.e., binary classification). We trained nine neural network models (Table 1). Because 75% of the SS dataset is labeled as empty, to avoid imbalance between the empty and nonempty classes, we took all

Fig. 4. While we train models on individual images, we only have labels for entire capture events (a set of images taken one after the other within approximately 1 second, e.g., *A*, *B*, and *C*), which we apply to all images in the event. When some images in an event have an animal (e.g., *A*) and others are empty (*B* and *C* in this example), the empty images are labeled with the animal type, which introduces some noise in the training-set labels and thus makes training harder.

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.

Table 1. Performance of different deep learning architectures

Architecture	No. of layers	Short description
AlexNet	8	A landmark architecture for deep learning winning ILSVRC 2012 challenge (31).
NiN	16	Network in Network (NiN) is one of the first architectures harnessing innovative 1 × 1 convolutions (49) to provide more combinational power to the features of a convolutional layers (49).
VGG	22	An architecture that is deeper (i.e., has more layers of neurons) and obtains better performance than AlexNet by using effective 3 × 3 convolutional filters (26).
GoogLeNet	32	This architecture is designed to be computationally efficient (using 12 times fewer parameters than AlexNet) while offering high accuracy (50).
ResNet	18, 34, 50, 101, 152	The winning architecture of the 2016 ImageNet competition (25). The number of layers for the ResNet architecture can be different. In this work, we try 18, 34, 50, 101, and 152 layers.

25% (757,000) nonempty images and randomly selected 757,000 empty images. This dataset was then split into training and test sets.

The training set contained 1.4 million images, and the test set contained 105,000 images. Since the SS dataset contains labels for only capture events (not individual images), we assigned the label of each capture event to all of the images in that event. All of the architectures achieved a classification accuracy of >95.8% on this task. The VGG model achieved the best accuracy of 96.8% (Table 2). To show the difficulty of the task and where the models currently fail, several examples for the best model (VGG) are shown in *SI Appendix*, *Results on the Volunteer-Labeled Test Set*, and *SI Appendix*, Fig. S10 shows the best model's confusion matrix.

Task II: Identifying Species. For this task, the corresponding output layer produced the probabilities of the input image being one of the 48 possible species. As is traditional in the field of computer vision, we reported top-1 accuracy (is the answer correct?) and top-5 accuracy (is the correct answer in the top-5 guesses by the network?). The latter is helpful in cases where multiple things appear in a picture, even if the ground-truth label in the dataset is only one of them. The top-5 score is also of particular interest in this work because AI can be used to help humans label data faster (as opposed to fully automating the task). In that context, a human can be shown an image and the AI's top-5 guesses. As we report below, our best techniques identified the correct animal in the top-5 list 99.1% of the time. Providing such a list thus could save humans the effort of finding the correct species name in a list of 48 species >99% of the time, although human-user studies will be required to test that hypothesis.

Measured on the expert-labeled test set, the model ensemble had 94.9% top-1 and 99.1% top-5 accuracy (*SI Appendix*, Fig. S11 shows its confusion matrix), while the best single model (ResNet-152) obtained 93.8% top-1 and 98.8% top-5 accuracy (Fig. 5, *Upper*). The results on the volunteer-labeled test set along with several examples (like Fig. 1) are reported in *SI Appendix*, *Results on the Volunteer-Labeled Test Set*.

Task III: Counting Animals. There are many different approaches for counting objects in images by deep learning (52–54), but nearly all of them require labels for bounding boxes around different objects in the image. Because this kind of information is not readily available in the SS dataset, we treated animal counting as a classification problem and left more advanced methods for future work. In other words, instead of actually counting animals in the image, we assigned the image to one of the 12 possible bins; each represented 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11–50, or +51 individuals, respectively. For this task, in addition to reporting top-1 accuracy, we also reported the percentage of images that were correctly classified within ±1 bin (1).

For this task, the ensemble of models on the expert-labeled test set got 63.1% top-1 accuracy, and 84.7% of predictions were within ±1 bin. *SI Appendix*, Fig. S10 shows the ensemble's confusion matrix. The same metrics for the best single model (ResNet-152) were 62.8% and 83.6%, respectively (Fig. 5, *Lower*). The results on the volunteer-labeled test set along with several examples are reported in *SI Appendix*, *Results on the Volunteer-Labeled Test Set*.

Task IV: Additional Attributes. The SS dataset contains labels for six additional attributes: standing, resting, moving, eating, interacting, and whether young are present (Fig. 1). Because these attributes are not mutually exclusive (especially for images containing multiple individuals), this task is a multilabel classification (55, 56) problem. A traditional approach for multilabel classification is to transform the task into a set of binary classification tasks (55, 57). We did so by having, for each additional attribute, one two-neuron softmax output layer that predicted the probability of that behavior existing (or not) in the image.

The expert-labeled test set does not contain labels for these additional attributes, so we used the majority vote among the volunteer labels as the ground truth label for each attribute. We counted an output correct if the prediction of the model for that attribute was >50% and matched the ground-truth label.

Table 2. Accuracy of different models on task I: Detecting images that contain animals

Architecture	Top-1 accuracy, %
AlexNet	95.8
NiN	96.0
VGG	**96.8**
GoogLeNet	96.3
ResNet-18	96.3
ResNet-34	96.2
ResNet-50	96.3
ResNet-101	96.1
ResNet-152	96.1
Ensemble of models	96.6

The bold font indicates the top-performing architecture.

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.

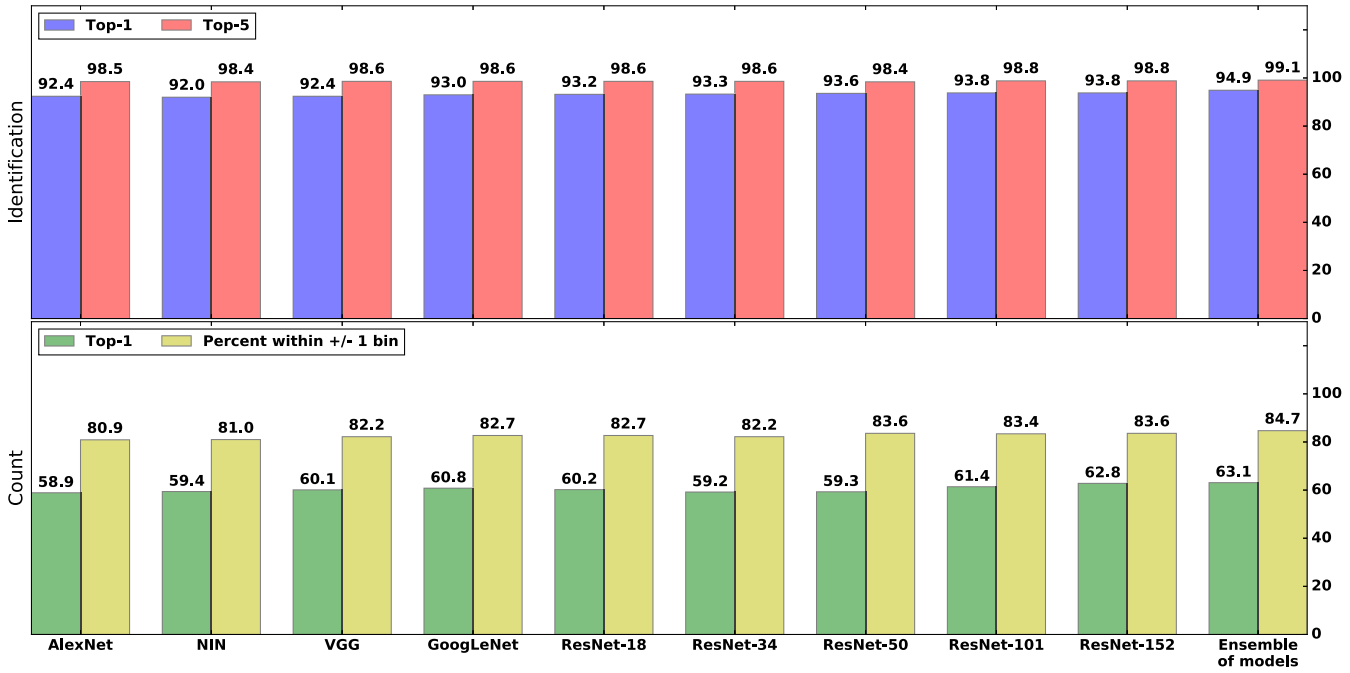

Fig. 5. (*Upper*) Top-1 and top-5 accuracy of different models on the task of identifying the species of animal present in the image. Although the accuracy of all of the models are similar, the ensemble of models is the best with 94.9% top-1 and 99.1% top-5 accuracy. (*Lower*) Top-1 accuracy and the percentage of predictions within ± 1 bin for counting animals in the images. Again, the ensemble of models is the best with 63.1% top-1 and 84.7% of the prediction within ± 1 bin.

We report traditional multilabel classification metrics, specifically, multilabel accuracy, precision, and recall (56). Pooled across all attributes, the ensemble of models produced 76.2% accuracy, 86.1% precision, and 81.1% recall. The same metrics for the best single model (ResNet-152) were 75.6%, 84.5%, and 80.9%, respectively. More results for predicting additional attributes are reported in *SI Appendix*, *Results on the Volunteer-Labeled Test Set*. For this and all previous tasks, we provide examples of correct predictions in Fig. 6 and incorrect network predictions in Fig. 7.

Saving Human Labor via Confidence Thresholding

One main benefit of automating information extraction is eliminating the need for humans to have to label images. Here, we estimated the total amount of human labor that could be saved if our system is designed to match the accuracy of human volunteers.

We created a two-stage pipeline by having the VGG model from the empty vs. animal experiment classify whether the image contained an animal and, if it did, having the ensemble of models from the second stage label it. We can ensure the entire pipeline is as accurate as human volunteers by having the network classify images only if it is sufficiently confident in its prediction.

Harnessing this confidence-thresholding mechanism, we can design a system that matches the volunteer human classification accuracy of 96.6%. For task I, detecting images that contain animals, we did not have expert-provided labels and thus did not know the accuracy of the human volunteers, so we assumed it to be the same 96.6% accuracy as on the animal identification task (task II). Because the VGG model's accuracy is higher than the volunteers', we can automatically process 75% of the data (because 75% of the images are empty) at human-level accuracy. For task II, identifying species, thresholding at 43% confidence enabled us to automatically process 97.2% of the remaining 25% of the data at human-level accuracy. Therefore, our fully automated system operated at 96.6% accuracy on $75\% \times 100\% + 25\% \times 97.2\% = 99.3\%$ of the data. Applying the same procedure to task III, counting animals, human volunteers were 90.0% accurate, and to match them, we thresholded at 79%. As a result, we can automatically count 44.5% of the nonempty images and therefore $75\% \times 100\% + 25\% \times 44.5\% = 86.1\%$ of the data. For more details and plots, refer to *SI Appendix*, *Confidence Thresholding*. We could not perform this exercise for task IV, additional attributes, because SS lacks expert-provided labels for this task, meaning human-volunteer accuracy on it is unknown.

Note that to manually label ~5.5 million images, nearly 30,000 SS volunteers have donated ~14.6 y of 40-h-a-week effort (1). Based on these statistics, our current automatic identification system would save an estimated 8.4 y of 40-h-per-week human labeling effort ($>$ 17,000 h) for 99.3% of the 3.2 million images in our dataset. Such effort could be reallocated to harder images or harder problems or might enable camera-trap projects that are not able to recruit as many volunteers as the famous SS project with its charismatic megafauna.

Helping Small Camera-Trap Projects via Transfer Learning

Deep learning works best with many (e.g., millions) labeled data points (here, images) (13). Many small camera-trap projects exist that do not have the ability to label a large set of images. Deep learning can still benefit such projects through transfer learning (32, 33, 58), wherein a network can first be trained on images available in other large datasets (e.g., large, public datasets like SS) and then further trained on a different, smaller dataset (e.g., a small camera-trap project with just a few thousand labeled images). The knowledge learned on the first dataset is thus repurposed to classify the second, smaller dataset. We conducted experiments validating this approach for the identifying-species task (tasks I and II), which gives a sense of how well smaller projects can expect to do with various amounts of labeled data.

Because we are not aware of any other publicly available labeled camera-trap datasets, to conduct this experiment, we simulated camera-trap projects of various sizes by randomly

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.

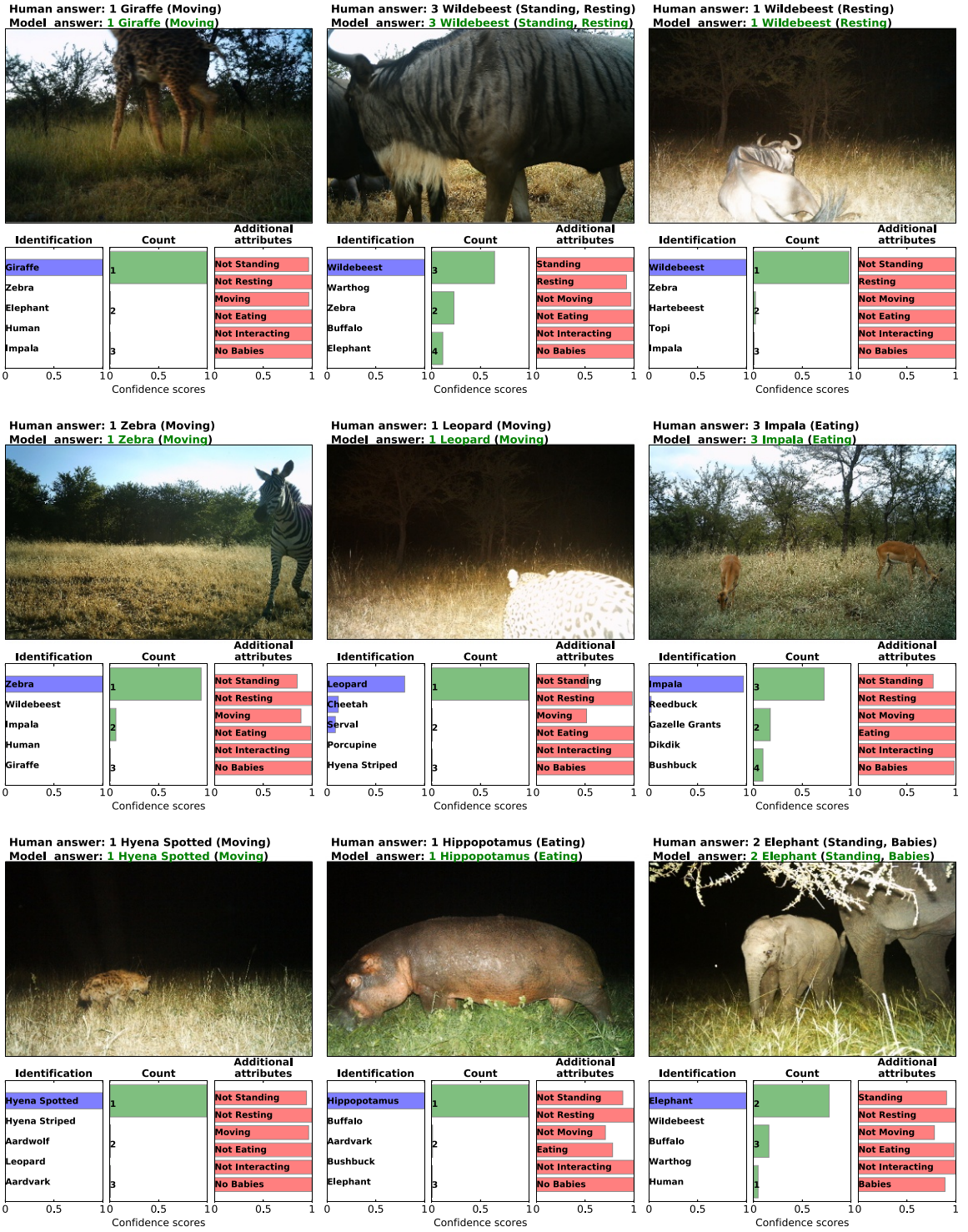

Fig. 6. Shown are nine images the ResNet-152 model labeled correctly. Above each image is a combination of expert-provided labels (for the species type and counts) and volunteer-provided labels (for additional attributes), as well as the model's prediction for that image. Below each image are the top guesses of the model for different tasks, with the width of the color bars indicating the model's output for each of the guesses, which can be interpreted as its confidence in that guess.

creating labeled datasets of different sizes from SS data. To conduct transfer learning, we first trained on the ImageNet dataset (59) and then further trained the network on a small simulated camera-trap dataset. ImageNet has 1.3 million labeled images for 1,000 categories (from synthetic objects such as bicycles and cars to wildlife categories such as dogs and lions). This dataset is commonly used in computer vision research, including research into transfer learning (32). Training on images from the real world can be helpful, even if the classes of images are dissimilar, because many lower-level image features (e.g., edge detectors of different orientations, textures, shapes, etc.) are common across very different types of images (32, 33, 58).

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.

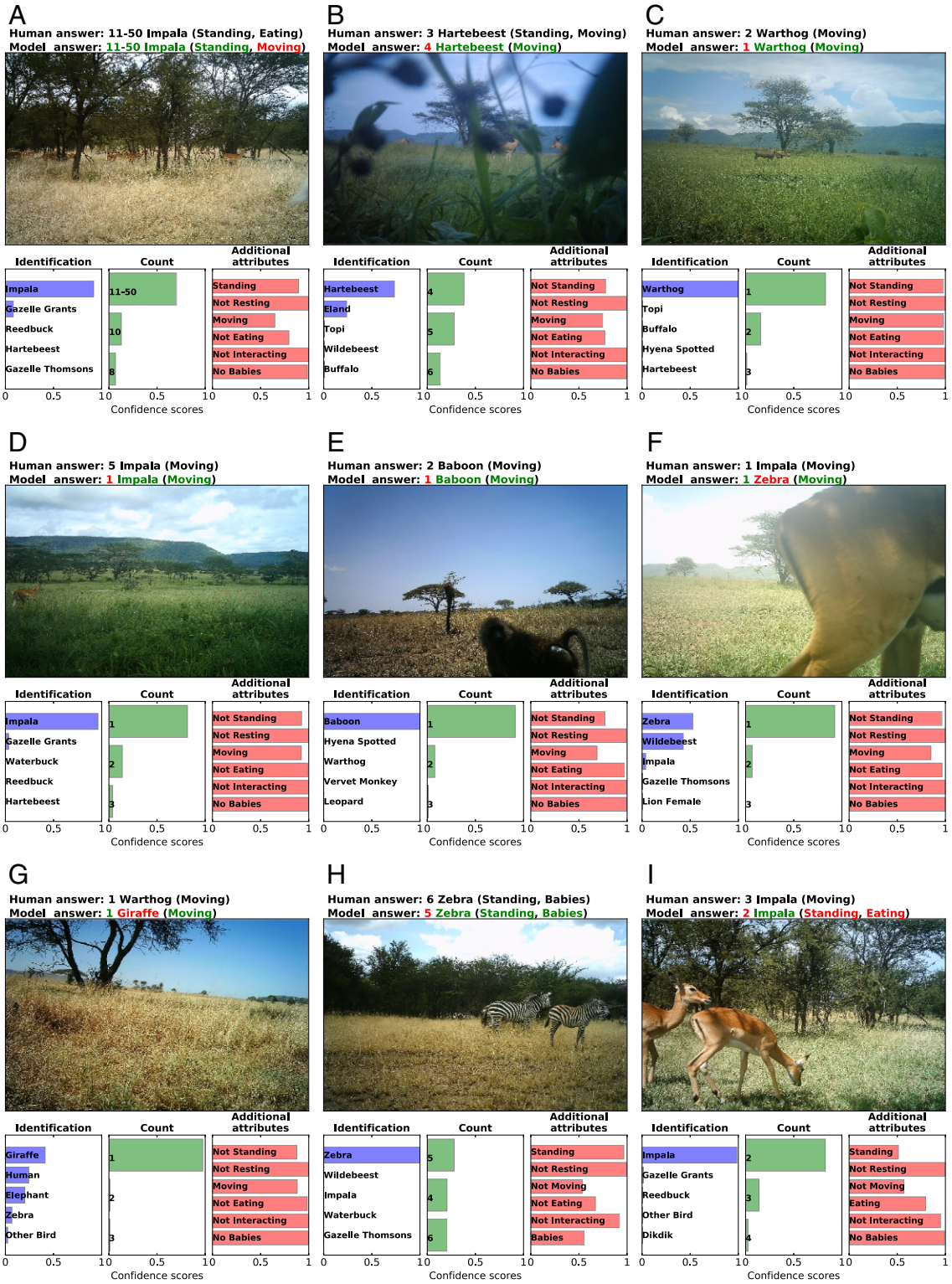

Fig. 7. (*A–I*) Shown are nine images the ResNet-152 model labeled incorrectly. Above each image are a combination of expert-provided labels (for the species type and counts) and volunteer-provided labels (for additional attributes), as well as the model's prediction for that image. Below each image are the top guesses of the model for different tasks, with the width of the color bars indicating the model's output for each of the guesses, which can be interpreted as its confidence in that guess. One can see why the images are difficult to get right. *G* and *I* contain examples of the noise caused by assigning the label for the capture event to all images in the event. *A*, *B*, *D*, and *H* show how animals being too far from the camera makes classification difficult.

That said, transfer learning from the ImageNet dataset to SS likely underestimates what performance is possible with transfer learning between camera-trap-specific datasets, because it has been shown that the more similar the classes of images are between the transfer-from and transfer-to datasets, the better transfer learning works (32). Transferring from the SS dataset to

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.

other wildlife camera-trap projects could thus provide even better performance. *SI Appendix, Transfer Learning* has additional details for these experiments.

The main takeaway is that a substantial fraction of the data can be automatically extracted at the same 96.6% accuracy level of citizen-scientists, even if only a few thousand labeled images are available. Accuracy, and thus automation percentages, further improves as more labeled data are provided during training. With 1.5 thousand (1.5k) images, >41% of the entire pipeline can be automated. Assuming a conservative 10 s per image, labeling these 1.5k images takes only 4.2 h. With only 3k images (8.3 h), that number jumps to >50%. With 6k, 10k, and 15k images (16.7, 27.8, and 41.7 h), 62.6%, 71.4%, and 83.0% of the data can be automatically labeled, respectively. With 50k images (138.9 h), 91.4% of the entire pipeline can be automated. Thus, sizable cost savings are available to small camera-trap projects of various sizes, and, especially at the low end, investing in labeling a few more thousand images can provide substantial performance improvements. *SI Appendix, Transfer Learning* provides full results, including more dataset sizes and the model's accuracy for task I, detecting images that contain animals, and task II, identifying species.

Discussion and Future Work

There are many directions for future work, but here we mention three particularly promising ones. The first is studying the actual time savings and effects on accuracy of a system hybridizing DNNs and teams of human volunteer labelers. Time savings should come from three sources: automatically filtering empty images, accepting automatically extracted information from images for which the network is highly confident in, and by providing human labelers with a sorted list of suggestions from the model so they can quickly select the correct species, counts, and descriptions. However, the actual gains seen in practice need to be quantified. Additionally, the effect of such a hybrid system on human accuracy needs to be studied. Accuracy could be hurt if humans are more likely to accept incorrect suggestions from DNNs, but could also be improved if the model suggests information that humans may not have thought to consider. A second, but related, promising direction is studying active learning (60, 61), a virtuous cycle in which humans label only the images in which the network is not confident, and then those images are added to the dataset, the network is retrained, and the process repeats. The third is automatically handling multispecies images, which we removed for simplicity. While our current trained pipeline can be applied to all images, for images with multiple species, it provides only one species label. In 97.5% of images, it correctly listed one of the species present, providing useful information, but the impact of missing the other species should be kept in mind and will depend on the use case. However, one could train networks to list multiple species via a variety of more sophisticated deep-learning techniques (47, 62, 63), a profitable area for future research.

Conclusions

In this work, we tested the ability of state-of-the-art computer vision methods called DNNs to automatically extract information from images in the SS dataset, the largest existing labeled dataset of wild animals. We first showed that DNNs can perform well on the SS dataset, although performance is worse for rare classes.

Perhaps most importantly, our results show that using deep-learning technology can save a tremendous amount of time for biology researchers and the human volunteers that help them by labeling images. In particular, for animal identification, our system can save 99.3% of the manual labor (>17,000 h) while performing at the same 96.6% accuracy level of human volunteers. This substantial amount of human labor can be redirected to other important scientific purposes and also makes knowledge extraction feasible for camera-trap projects that cannot recruit large armies of human volunteers. Automating data extraction can thus dramatically reduce the cost to gather valuable information from wild habitats and will thus likely enable, catalyze, and improve many future studies of animal behavior, ecosystem dynamics, and wildlife conservation.

ACKNOWLEDGMENTS. We thank Sarah Benson-Amram, the SS volunteers, and the members of the Evolving AI Laboratory at the University of Wyoming for valuable feedback, especially Joost Huizinga, Tyler Jaszkowiak, Roby Velez, Henok Mengistu, and Nick Cheney. J.C. was supported by National Science Foundation CAREER Award 1453549.

1. Swanson A, et al. (2015) Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Sci Data* 2:150026.
2. Harris G, Thompson R, Childs JL, Sanderson JG (2010) Automatic storage and analysis of camera trap data. *Bull Ecol Soc Am* 91:352–360.
3. O'Connell AF, Nichols JD, Karanth KU (2010) *Camera Traps in Animal Ecology: Methods and Analyses* (Springer, Tokyo).
4. Silveira L, Jacomo AT, Diniz-Filho JAF (2003) Camera trap, line transect census and track surveys: A comparative evaluation. *Biol Conserv* 114:351–355.
5. Bowkett AE, Rovero F, Marshall AR (2008) The use of camera-trap data to model habitat use by antelope species in the Udzungwa mountain forests, Tanzania. *Afr J Ecol* 46:479–487.
6. Fegraus EH, et al. (2011) Data acquisition and management software for camera trap data: A case study from the team network. *Ecol Inform* 6:345–353.
7. Krishnappa YS, Turner WC (2014) Software for minimalistic data management in large camera trap studies. *Ecol Inform* 24:11–16.
8. Swinnen KRR, Reijniers J, Breno M, Leirs H (2014) A novel method to reduce time investment when processing videos from camera trap studies. *PLoS One* 9:e98881.
9. Swanson A, Arnold T, Kosmala M, Forester J, Packer C (2016) In the absence of a "landscape of fear": How lions, hyenas, and cheetahs coexist. *Ecol Evol* 6:8534–8545.
10. Palmer MS, Fieberg J, Swanson A, Kosmala M, Packer C (2017) A 'dynamic' landscape of fear: Prey responses to spatiotemporal variations in predation risk across the lunar cycle. *Ecol Lett* 20:1364–1373.
11. Anderson TM, et al. (2016) The spatial distribution of African savannah herbivores: Species associations and habitat occupancy in a landscape context. *Phil Trans R Soc B* 371:20150314.
12. Palmer MS, Packer C (2018) Giraffe bed and breakfast: Camera traps reveal Tanzanian yellow-billed oxpeckers roosting on their large mammalian hosts. *Afr J Ecol*.
13. Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning* (MIT Press, Cambridge, MA).
14. Swanson A, Kosmala M, Lintott C, Packer C (2016) A generalized approach for producing, quantifying, and validating citizen science data from wildlife images. *Conserv Biol* 30:520–531.
15. Samuel AL (1959) Some studies in machine learning using the game of checkers. *IBM J Res Dev* 3:210–229.
16. Mohri M, Rostamizadeh A, Talwalkar A (2012) *Foundations of Machine Learning* (MIT Press, Cambridge, MA).
17. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444.
18. Hu W, Huang Y, Wei L, Zhang F, Li H (2015) Deep convolutional neural networks for hyperspectral image classification. *J Sensors* 2015:1–10.
19. Bridle JS (1990) Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. *Neurocomputing* (Springer, New York), pp 227–236.
20. Hinton G, et al. (2012) Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Proc Mag* 29:82–97.
21. Deng L, Hinton G, Kingsbury B (2013) New types of deep neural network learning for speech recognition and related applications: An overview. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, New York).
22. Bahdanau D, et al. (2016) End-to-end attention-based large vocabulary speech recognition. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, New York).
23. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. *2014 Advances in Neural Information Processing Systems (NIPS)* (Neural Information Processing Systems Foundation, La Jolla, CA).
24. Cho K, et al. (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078.
25. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York).
26. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.
27. Mnih V, et al. (2015) Human-level control through deep reinforcement learning. *Nature* 518:529–533.
28. Figueroa K, Camarena-Ibarrola A, García J, Villela HT (2014) Fast automatic detection of wildlife in images from trap cameras. *Progress in Pattern Recognition, Image*

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.

Analysis, Computer Vision, and Applications: 19th Iberoamerican Congress, eds Bayro-Corrochano E, Hancock E (Springer International Publishing, Cham, Switzerland), pp 940–947.
29. Yu X, et al. (2013) Automated identification of animal species in camera trap images. *EURASIP J Image Vide* 2013:52.
30. Chen G, Han TX, He Z, Kays R, Forrester T (2014) Deep convolutional neural network based species recognition for wild animal monitoring. *2014 IEEE International Conference on Image Processing (ICIP)* (IEEE, New York).
31. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *2012 Advances in Neural Information Processing Systems (NIPS)* (Neural Information Processing Systems Foundation, La Jolla, CA).
32. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? *2014 Advances in Neural Information Processing Systems (NIPS)* (Neural Information Processing Systems Foundation, La Jolla, CA).
33. Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. *IEEE T Pattern Anal* 35:1798–1828.
34. Graves A, Mohamed Ar, Hinton G (2013) Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, New York).
35. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv:1409.0473.
36. Wang H, et al. (2014) Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features. *J Med Imaging* 1:034003.
37. Rampasek L, Goldenberg A (2018) Learning from everyday images enables expert-like diagnosis of retinal diseases. *Cell* 172:893–895.
38. Kashif MN, Raza SEA, Sirinukunwattana K, Arif M, Rajpoot N (2016) Handcrafted features with convolutional neural networks for detection of tumor cells in histology images. *2016 IEEE 13th International Symposium on Biomedical Imaging* (IEEE, New York), pp 1029–1032.
39. Chherawala Y, Roy PP, Cheriet M (2013) Feature design for offline Arabic handwriting recognition: Handcrafted vs. automated? *2013 International Conference on Document Analysis and Recognition (ICDAR)* (IEEE, New York).
40. Park SR, et al. (2018) De-multiplexing vortex modes in optical communications using transport-based pattern recognition. *Opt Express* 26:4004–4022.
41. Yang J, Yu K, Gong Y, Huang T (2009) Linear spatial pyramid matching using sparse coding for image classification. *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York).
42. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. *J Mach Learn Res* 3:993–1022.
43. Fei-Fei L, Perona P (2005) A Bayesian hierarchical model for learning natural scene categories. *2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York), 2:524–531.
44. 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. *2016 International Symposium on Visual Computing* (Springer, Cham, Switzerland), pp 747–756.
45. Gomez A, Salazar A, Vargas F (2016) Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. arXiv:1603.06169v2.
46. Deng J, et al. (2009) Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York).
47. Caruana R (1998) Multitask learning. *Learning to Learn* (Springer, New York), pp 95–133.
48. Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. *2008 International Conference on Machine Learning (ICML)* (Association for Computing Machinery, New York).
49. Lin M, Chen Q, Yan S (2013) Network in network. arXiv:1312.4400.
50. Szegedy C, et al. (2015) Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York).
51. Dauphin YN, et al. (2014) Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *2014 Advances in Neural Information Processing Systems (NIPS)* (Neural Information Processing Systems Foundation, La Jolla, CA).
52. Chattopadhyay P, Vedantam R, Ramprasaath R, Batra D, Parikh D (2016) Counting everyday objects in everyday scenes. CoRR, abs/1604.03505 1:10.
53. Onoro-Rubio D, López-Sastre RJ (2016) Towards perspective-free object counting with deep learning. *2016 European Conference on Computer Vision (ECCV)*.
54. Zhang C, Li H, Wang X, Yang X (2015) Cross-scene crowd counting via deep convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York).
55. Tsoumakas G, Katakis I (2006) Multi-label classification: An overview. *Int J Data Warehous* 3:1–13.
56. Sorower MS (2010) *A Literature Survey on Algorithms for Multi-Label Learning* (Oregon State University, Corvallis, OR), Vol 18.
57. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. *Mach Learn* 85:333–359.
58. Donahue J, et al. (2014) Decaf: A deep convolutional activation feature for generic visual recognition. *2014 International Conference on Machine Learning (ICML)* (Association for Computing Machinery, New York).
59. Russakovsky O, et al. (2015) Imagenet large scale visual recognition challenge. *Int J Comput Vis* 115:211–252.
60. Settles B (2012) Active learning. *Synth Lectures Artif Intelligence Machine Learn* 6:1–114.
61. Sener O, Savarese S (2018) Active learning for convolutional neural networks: A core-set approach. *International Conference on Learning Representations*. Available at https://openreview.net/forum?id=H1aluk-RW. Accessed May 25, 2018.
62. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. *2015 Advances in Neural Information Processing Systems (NIPS)* (Neural Information Processing Systems Foundation, La Jolla, CA).
63. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York).

Downloaded from https://www.pnas.org by UNIVERSITEIT VAN AMSTERDAM (UVA) 215 on February 20, 2024 from IP address 145.3.24.174.