Vélez, J, McShea, W, Shamon, H, Castiblanco-Camacho, PJ, Tabak, MA, Chalmers, C, Fergus, P and Fieberg, J

An evaluation of platforms for processing camera-trap data using artificial intelligence

Received: 6 February 2022 | Accepted: 18 November 2022

DOI: 10.1111/2041-210X.14044

Methods in Ecology and Evolution | BRITISH ECOLOGICAL SOCIETY

REVIEW

An evaluation of platforms for processing camera-trap data using artificial intelligence

Juliana Vélez[1,2] | William McShea[2] | Hila Shamon[2] | Paula J. Castiblanco-Camacho[3] | Michael A. Tabak[4] | Carl Chalmers[5] | Paul Fergus[5] | John Fieberg[1]

[1]Department of Fisheries, Wildlife and Conservation Biology, University of Minnesota, Saint Paul, Minnesota, USA

[2]Conservation Ecology Center, Smithsonian's National Zoo and Conservation Biology Institute, Front Royal, Virginia, USA

[3]Departamento de Ciencias Biológicas, Universidad de los Andes, Bogotá, Colombia

[4]Quantitative Science Consulting, LLC, Toronto, Ontario, Canada

[5]School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, UK

Correspondence
Juliana Vélez
Email: julianavelezgomez@gmail.com

Funding information
Colciencias - Fulbright Colombia; Department of Fisheries, Wildlife and Conservation Biology, University of Minnesota; Interdisciplinary Center for the Study of Global Change; Smithsonian's National Zoo and Conservation Biology Institute; WWF's Russell E. Train Education for Nature Program (EFN)

Handling Editor: Res Altwegg

Abstract

1. Camera traps have quickly transformed the way in which many ecologists study the distribution of wildlife species, their activity patterns and interactions among members of the same ecological community. Although they provide a cost-effective method for monitoring multiple species over large spatial and temporal scales, the time required to process the data can limit the efficiency of camera-trap surveys. Thus, there has been considerable attention given to the use of artificial intelligence (AI), specifically deep learning, to help process camera-trap data. Using deep learning for these applications involves training algorithms, such as convolutional neural networks (CNNs), to use particular features in the camera-trap images to automatically detect objects (e.g. animals, humans, vehicles) and to classify species.
2. To help overcome the technical challenges associated with training CNNs, several research communities have recently developed platforms that incorporate deep learning in easy-to-use interfaces. We review key characteristics of four AI platforms—Conservation AI, MegaDetector, MLWIC2: Machine Learning for Wildlife Image Classification and Wildlife Insights—and two auxiliary platforms—Camelot and Timelapse—that incorporate AI output for processing camera-trap data. We compare their software and programming requirements, AI features, data management tools and output format. We also provide R code and data from our own work to demonstrate how users can evaluate model performance.
3. We found that species classifications from Conservation AI, MLWIC2 and Wildlife Insights generally had low to moderate recall. Yet, the precision for some species and higher taxonomic groups was high, and MegaDetector and MLWIC2 had high precision and recall when classifying images as either 'blank' or 'animal'. These results suggest that most users will need to review AI predictions, but that AI platforms can improve efficiency of camera-trap-data processing by allowing users to filter their dataset into subsets (e.g. of certain taxonomic groups or blanks) that can be verified using bulk actions.

4. By reviewing features of popular AI-powered platforms and sharing an open-source GitBook that illustrates how to manage AI output to evaluate model performance, we hope to facilitate ecologists' use of AI to process camera-trap data.

KEYWORDS
artificial intelligence, camera traps, computer vision, data processing, deep learning, image classification, remote sensing, review

1 | INTRODUCTION

Camera traps are frequently used in ecological research to study animal behaviour and to estimate density, relative abundance or occupancy in single- and multiple-species studies (Burton et al., 2015). Camera traps can generate tremendous amounts of image data, and thus, much attention has been given recently to developing artificial intelligence (AI) approaches for processing images using deep learning algorithms. These algorithms can perform image classification and object detection after being trained using a pre-labelled dataset that uniquely identifies each species (or category) of interest. AI has been widely used for removing empty images (i.e. images without animals, also referred to blanks; Beery et al., 2018), species identification (Carl et al., 2020; Gomez Villa et al., 2017; Norouzzadeh et al., 2018; Schneider et al., 2018; Tabak et al., 2018; Whytock et al., 2021), counting of individuals when there is a single species in an image (Norouzzadeh et al., 2018) and individual recognition of animals present in the training dataset (Bogucki et al., 2018; Chen et al., 2020; Schneider, Taylor, et al., 2020). Others have reviewed and compared the performance of different state-of-the-art classification methods and deep learning architectures for identifying species in camera-trap images (Norouzzadeh et al., 2018; Schneider et al., 2018) and videos (Chen et al., 2019).

Although AI makes it possible to process millions of images in short time periods (e.g. 1 million images in 24 h), large and diverse amounts of pre-processed data may be required to train models. In addition, the performance of AI approaches may suffer when models are developed using unbalanced training datasets (e.g. with highly variable numbers of images of each species; Gomez Villa et al., 2017), small and geographically limited datasets but then applying the model more broadly (Beery et al., 2018; Schneider, Greenberg, et al., 2020; Tabak et al., 2018), or when applying the model to low-resolution images (although see Gomez et al., 2016 for strategies to improve recognition of poor-quality images using deep learning). In addition, model creation and refinement require technical and programming expertise beyond the limits of many ecologists (Christin et al., 2019; Tabak et al., 2020). For example, specialized techniques may be needed to increase the number of images of rare species in the training dataset. Augmentation of the training data can be performed by simulating animals on empty images and modifying features such as animal pose, illumination and orientation (Beery et al., 2020). Other alternatives of training augmentation include the re-sampling of images based on a stochastic method that includes rare classes more frequently (Schneider, Greenberg, et al., 2020). It can also be useful to identify particular species or sites where models perform poorly, and then use data from those species or sites to further train available models (Tabak et al., 2020).

To reach a wider audience of camera-trap users, several initiatives have recently been launched with the goal of training AI models with broad and diverse image datasets and creating platforms that facilitate the use of AI via simple user interfaces and software. Examples of these initiatives include Camelot (Hendry & Mann, 2018), Conservation AI (Chalmers et al., 2019), MegaDetector (Beery et al., 2019), MLWIC2: Machine Learning for Wildlife Image Classification (Tabak et al., 2020), Timelapse (Greenberg et al., 2019) and Wildlife Insights (Ahumada et al., 2020). These platforms differ in several aspects including their ease of use, required computer and programming skills, data management tools and whether they focus only on coarse categorization of images or include the ability to classify species. Thus, platforms may be more or less suitable, depending on the user's needs and abilities.

In addition to providing access to trained AI models, AI-powered platforms can enable users to integrate AI output with standard camera-trap-processing workflows. For example, users might want to record additional image information not targeted by the AI model, including specific animal features (e.g. age, sex, stripe or spot patterns). Uniquely identifying characteristics, for instance, may allow estimation of species density or abundance using spatial capture–recapture methods (Augustine et al., 2018; Efford & Fewster, 2013; Royle et al., 2013). Other specific animal features, such as animal health characteristics, group sizes or animal behaviour might also be of interest, as well as environmental conditions or signs of human activity within the camera's field of view (Greenberg et al., 2019; Norouzzadeh et al., 2018).

Greenberg (2020) discussed important aspects that need to be considered before using AI for automated image recognition, including knowing characteristics of the training dataset (e.g. species included, number of images per species and geographical locations of the image data). In addition, he emphasized the need to use human verification to account for errors in AI output and provided a series of recommendations for processing camera-trap data using AI. Specifically, he recommended that users filter images with high confidence values associated with their AI predictions, and then review these images using bulk actions (e.g. selecting multiple species and accepting AI labels or correcting wrong labels provided by AI). It is important to note, however, that although higher confidence values are generally associated with more accurate predictions, confidence values do not provide an accurate measure of predictive uncertainty, and high confidence values do not guarantee correct AI classifications (Guo et al., 2017).

We build on this prior work by providing an overview of some of the AI-powered platforms currently available to the public, discuss how AI output can be integrated to process camera- trap data, and provide a detailed walkthrough of using and evaluating each platform in an associated online Gitbook (Vélez & Fieberg, 2022). In Section 2, we compare fully-automated and semi-automated image processing workflows. In Section 3, we review the features of different AI-powered platforms for data upload, image identification, model training and post-processing of classified images. We consider platforms with diverse characteristics to illustrate a wide range of options and based on our perception of their stability and developer responsiveness. In Section 4, we evaluate the performance of AI platforms for animal detection and species classification using four out-of-sample datasets representing environments in Africa, Asia, North America and South America. Finally, we discuss the implications of our findings for users looking to incorporate an AI platform into a workflow for processing camera-trap data. We provide a more detailed overview of each AI platform and code for evaluating AI performance through an open-source GitBook (Vélez & Fieberg, 2022).

2 | WORKFLOWS: FULLY-AUTOMATED VS. SEMI-AUTOMATED RECOGNITION

Fully automated recognition refers to pipelines in which *computer vision* is used for detecting and identifying species or features in images without human review. A fully automated workflow is particularly useful for projects that require near-real-time detection (e.g. for preventing human–wildlife conflict), long-term projects with limited human capacity for image processing, projects with multiple deployments in the same geographical region, and projects that do not require further data annotation by humans to record information not captured by the AI model.

A fully automated recognition workflow requires a trained model capable of identifying all classes of interest and providing highly accurate classifications. Users that desire a fully automated workflow will likely need to leverage data collected from their specific area, which may require training their own models, or using similar area-specific models trained by others. This will ensure the model includes the species of interest and that classifications are accurate. Accuracy is often degraded when previously trained models are applied to new camera locations as background scenes can differ (Schneider, Greenberg, et al., 2020). However, accuracy for new camera locations can be further increased by retraining models using images from those new camera locations. Users should also be aware that model performance may vary by species, and the impact of mis-classifications will depend on the underlying objectives, analysis approach and target of estimation (Schneider, Greenberg, et al., 2020; Whytock et al., 2021).

Although a fully automated workflow sounds appealing, some models might not reach an accuracy level that fulfils a user's needs. Instead, a semi-automated workflow can be implemented by integrating computer vision with *human vision* to facilitate image processing. Semi-automated workflows may facilitate image review by using AI output to filter and group images by categories that can be easily inspected (Greenberg, 2020). For example, empty images or images containing particular species with high confidence values associated with their predictions (i.e. images with a high probability of being correctly labelled by the model) can be filtered and quickly reviewed and verified using batch image selection. Some platforms allow the user to interactively change the confidence threshold (i.e. the confidence limit used to accept AI predictions) when selecting and filtering data (Greenberg et al., 2019). Inspecting model performance across a range of confidence thresholds is advisable to determine an appropriate threshold for batch processing. Another common feature provided by some platforms is the display of bounding boxes around detected animals, which can be particularly useful for locating small mammals and birds.

3 | WHICH PLATFORM SHOULD I USE?

An initial determining factor in selecting an appropriate platform is whether users have data that can be made public. Some platforms, such as MegaDetector and MLWIC2, were developed to maintain private workflows, while others, such as Wildlife Insights, are oriented towards open data and public data repositories. In addition, platforms differ in their ease of use, and a user's operating system and internet access may also play a role in determining an appropriate platform. Another important consideration is whether users only need to discriminate between blanks and images with an animal or whether they need accurate species classifications. Because it can be difficult to achieve high accuracy rates when existing models are applied to novel data and environments (Schneider, Greenberg, et al., 2020), users will typically want to select a platform that allows them to easily review images (using bulk selection/verification of images and image sorting/filtering) along with AI output so they can correct mis-classified images.

In addition to AI performance, users should consider different steps in a standard camera-trap data-processing workflow when choosing a platform. These include (1) data organization and management, (2) image annotation (e.g. species/individual classification, or tagging of additional information), (3) image data extraction (i.e. extraction of image metadata, image labels and other annotations), (4) data exploration (i.e. summary statistics, data analyses) and data export (i.e. output files for subsequent analyses) (Greenberg et al., 2019; McShea et al., 2016; Niedballa et al., 2016; Young et al., 2018). We compare AI-powered platforms that perform species classifications (Conservation AI, MLWIC2: Machine Learning for Wildlife Image Classification and Wildlife Insights), and animal detection (MegaDetector, MLWIC2: Machine Learning for Wildlife Image Classification), as well as auxiliary platforms that integrate MegaDetector output for image processing (Camelot and Timelapse). To compare platforms, we consider an extensive set of criteria that includes features found in standard camera-trap processing workflows and related to the AI modules (Tables 1 and 2).

3.1 | Wildlife Insights

Wildlife Insights is an initiative developed by a partnership between Conservation International, Wildlife Conservation Society, World Wildlife Fund, Zoological Society of London, the Smithsonian Institution, North Carolina Museum of Natural Sciences, Yale University and Google (Ahumada et al., 2020). Wildlife Insights serves as a data library and data-sharing platform in the cloud. Users can upload labelled or unlabelled images through a Web-based upload tool, an application programming interface or a desktop client. To promote data sharing and research collaboration addressing ecological questions at regional or global scales (e.g. assessment of species declines in response to climate change), Wildlife Insights requires verified users to share their data under a Creative Commons licence (CC0, CC BY 4.0, or CC BY-NC 4.0) after a maximum embargo period of 48 months. Other users can download data from the image repository using filters provided by the interface (e.g. to select for particular species, regions, dates). Public downloads will not contain exact coordinates of records of threatened terrestrial vertebrates (Critically Endangered (CR), Endangered (EN) or Vulnerable (VU) based on the IUCN Red List), to prevent exposure of geographical location of species that might be at risk (https://www.wildlifeinsights.org/sensitive-species, accessed on 27/06/2022). Wildlife Insights also provides tools for using AI to detect blank images and to identify over 993 different animal species from around the world (https://www.wildlifeinsights.org/about-wildlife-insights-ai, accessed on 27/06/2022). Wildlife Insights uses a model trained using EfficientNet Convolutional Neural Networks for image classification with labelled camera-trap images collected by Wildlife Insights partners at sites worldwide. Wildlife Insights also provides bounding boxes in the interface, which is powered by a custom object-detection model. While uploading images to Wildlife Insights, they will be processed using AI, and then the user can download the resulting species classifications and metadata (e.g. time and date), which is automatically extracted by the system (Ahumada et al., 2020). Users can organize images hierarchically (e.g. by projects, sub-projects and deployments), and therefore Wildlife Insights can serve as a project management tool. Wildlife Insights includes an interface to facilitate image processing and verification of model classifications and to allow users to annotate images with additional information not targeted by AI. Images can be processed in bursts (i.e. by grouping images within a time frame), and the cloud-based infrastructure makes it easy for multiple collaborators to process images simultaneously. Wildlife Insights also includes an analysis module that provides various data summaries including species records, sites surveyed and sampling periods.

Pros: Data organization/management system, features for image review/annotation and multi-user environments; serves as an image repository and provides advanced reporting and analytical capabilities; tools for validating AI predictions within the platform; no programming expertise is required to run AI models; AI models for species classification were trained on a global scale.

Cons: Mandatory data sharing after an embargo period; cloud-based, which makes it susceptible to connection instability and service outages; does not include tools for model training by individual users.

3.2 | MegaDetector

MegaDetector is a model trained using global data to detect blanks, animals, people and vehicles from camera-trap images (Beery et al., 2019). The MDv5 model is based on the YOLOv5 architecture and is hosted in the Microsoft/CameraTraps GitHub repository, where it can be downloaded by users that want to run the model on their own (Beery et al., 2019). The MDv5 version increased the running speed compared to the previous version (3–4× faster than MDv4.1), and can process around 40,000 images per day when using a standard computer and around 800,000 images when using a Graphics Processing Unit (which performs efficient computations by doing them in parallel). To run MegaDetector, users will need to be comfortable running computer code at the command line. Alternatively, users can contact MegaDetector developers who will then run the model for them once their data are transferred. Although data will need to be visible to developers during processing in this scenario, they will not be shared or released publicly.

MegaDetector provides a JSON file as output, which indicates the locations of detected objects in each image and associated confidence values for each detection. Users can run a Python script to sort, move and organize images according to the MegaDetector predictions. When performing this task, users can choose a specific confidence threshold for determining which classifications should be accepted versus considered blank (e.g. using a 0.25 confidence threshold would classify images with confidence values less than 0.25 as 'Blank'). MegaDetector can also facilitate an extra post-processing step to reduce false positives (e.g. due to vegetation or background features that MegaDetector identifies as an animal), thereby increasing model accuracy. This step, implemented using a Python script (see Microsoft/CameraTraps/api/batch/postprocessing/), involves identifying detections that have exactly the same bounding box across many images. Users can further process MegaDetector output using other platforms, such as Camelot, Timelapse and Zooniverse, as part of a semi-automated workflow to classify species.

TABLE 1 Overview of the requirements, artificial intelligence (AI) modules, and tools for data management, image processing, and data exploration associated with four popular AI platforms, Conservation AI, MegaDetector, MLWIC2 and Wildlife Insights

Workflow criteria	Features	Conservation AI	MegaDetector	MLWIC2	Wildlife insights
Overview	Description	Cloud-based platform that uses AI for classification of species in camera-trap images, videos, acoustic recordings and drone images	AI model for classification of animals, people and vehicles in camera trap images. Its output can be integrated with different platforms (e.g. Camelot, Timelapse or Zooniverse) to facilitate camera-trap image processing	R package and Shiny Application that uses AI for classification of species in camera-trap images	Cloud-based platform for management and processing of camera-trap images. Wildlife Insights uses AI to provide species classifications to facilitate camera-trap image processing
	Developers	Liverpool John Moores University (UK)	Sara Beery, Dan Morris and Siyu Yang	Michael A. Tabak—Quantitative Science Consulting, LLC, Ontario, Canada	Conservation International, Wildlife Conservation Society, World Wildlife Fund, Zoological Society of London, the Smithsonian Institution, North Carolina Museum of Natural Sciences, Yale University and Google
Requirements	Cost	Free	Free	Free	Charges may apply to large NGOs, government agencies and companies; evaluated on a case-by-case basis
	Public data or shared with third party during analysis	Images will be visible to developers at Conservation AI during processing. After processing, images are made public to users with a Conservation AI account	If users receive assistance from MegaDetector developers to run the model, data will need to be visible to them during processing but will not be shared or released publicly	No	Wildlife Insights requires users to share their data under a Creative Commons licence (CC0, CC BY 4.0, or CC BY-NC 4.0) after a maximum embargo period of 48 months
	Internet access required	All functionality in the cloud	For downloading prerequisites and MegaDetector for local use. For data transfer if MegaDetector developers run the model for the user	For downloading prerequisites and MLWIC2 model	All functionality in the cloud
	Prerequisites/ software	Conservation AI account	For local run, installation of Anaconda, Git and NVIDIA driver (if using a GPU)	Users need to install R, Anaconda Navigator, Python (3.5, 3.6 or 3.7), Rtools (for Windows computers), version 1.14 of TensorFlow and the MLWIC2 R package	Verified Wildlife Insights account
	Operating system/ hardware supported	Any, as it is cloud based	Windows, macOS and Linux. GPU use supported	Windows, macOS. GPU use supported	Any, as it is cloud based

(Continues)

TABLE 1 (Continued)

Workflow criteria	Features	Conservation AI	MegaDetector	MLWIC2	Wildlife insights
	Data format	Images: JPG, PNG. Resolution no greater than 2000 × 2000 pixels; Videos: MP4	JPG, PNG	JPG, PNG, PDF	JPG, PNG, HEIC
	Memory/storage capacity	Unlimited	Unlimited. Based on local memory/storage capacity if run locally	Unlimited. Based on local memory/storage capacity	Unlimited
	Data upload/import options	Images can be shared with developers through cloud-based storage (e.g. Google Drive) or directly uploaded to the platform using a batch upload of up to 1,000 images at a time	For assisted model runs, the MegaDetector developers will provide instructions for bulk data transfer	Runs locally, does not require data upload. MLWIC2 provides functions to format filenames prior to running the AI model	Images are imported to Wildlife Insights by browsing files on a local computer. Manual or CSV bulk import of information about projects, cameras and deployments, including geographical coordinates associated with each camera and the dates it was in operation
AI module	Classification categories and training region	Models available identify humans, man-made objects (e.g. cars and fires) and species from the United Kingdom, South Africa, North America and Tanzania	MegaDetector is a model trained to detect animals, people and vehicles from camera-trap images. Detects multiple objects in an image. Training performed using images collected at sites worldwide	The 'species_model' classifies 58 species from North America. The 'empty_animal' model classifies 'Blank' and 'Animal' categories. Training performed using images from 10 states across USA	Wildlife Insights model detects blank images and identifies over 993 different animal species from around the world. Training performed using images collected at sites worldwide
	How to use AI	Once uploaded to the platform, images are automatically classified using the available AI models	MegaDetector can be run: (1) Locally by running Python code at the command line. (2) Assisted by MegaDetector developers after data are transferred. (3) Using a batch processing Application Programming Interface (recommended for large datasets with millions of images). (4) Through Camelot or Zooniverse	Users must know the R language and be familiar with file path specifications. MLWIC2 can be used either through an R programming console or a graphical user interface (Shiny Application)	While uploading, images are classified using the Wildlife Insights model
	AI model	Deep convolutional neural network based on Faster RCNN Resnet 101	MDv5 based on the YOLOv5 object-detection architecture and hosted in the Microsoft/CameraTraps GitHub repository	Deep convolutional neural network based on a ResNet-18 architecture	Model trained using EfficientNet Convolutional Neural Networks for image classification

TABLE 1 (Continued)

Workflow criteria	Features	Conservation AI	MegaDetector	MLWIC2	Wildlife insights
	Processing time	Around 10,000 images per hour	MDv5 can process around 40,000 images per day when using a standard computer and around 800,000 images per day when using a Graphics Processing Unit	Around 20,000 images per hour using a standard computer without a Graphics Processing Unit	On a single CPU, the AI model can process around 18,000 images per hour
	Near-real-time data	It provides services for image detection and classification in near-real time from linked devices capable of transferring images using a Simple Mail Transfer Protocol (SMTP)	No	No	No
	Bounding boxes	Yes	Yes	No	Yes
	Interface for AI output validation	To visualize images, users can filter by confidence threshold, survey or trap	Users can validate MegaDetector output via integration with Camelot, Timelapse or Zooniverse	No	To review and annotate images, users can filter by species, survey, trap or image status ('Blank' vs. 'Not Blank')
	Model training available	Conservation AI also provides a module for image tagging and model training for specific datasets. Users need to upload images into the tagging module (up to 500 images at a time), where they will create the training dataset	No	The R package and Shiny Application provide functionality to train your own model using labelled images	No
Data management	Data input schema	—	—	—	Wildlife Insights uses a default data input schema to record information from images, including species classification, animal count, age, sex, markings and individual ID
	Image repository	—	—	—	Data library and data-sharing platform in the cloud. Other users can download data from the image repository using filters provided by the interface (e.g. to select for particular species, regions, dates)

(Continues)

TABLE 1 (Continued)

Workflow criteria	Features	Conservation AI	MegaDetector	MLWIC2	Wildlife insights
	Data management system (e.g. data organization by projects, deployments, locations)	—	—	—	Data can be organized by projects, initiatives, subprojects and deployments. Images will be presented in the Wildlife Insights interface that serves as a dashboard to visualize and annotate images, selecting one or multiple images at a time
Image processing tools	Features for image editing (e.g. zooming, brightness and contrast) and image/metadata review	—	—	—	Image editing (brightness, contrast and saturation), and image metadata inspection
	Multi-user environments	Multiple users can be associated with the same project to compile the training set in the tagging module	—	—	Project owners can give access to collaborators with a Wildlife Insights account and select their role within the project (e.g. tagger, editor, viewer)
	Bulk actions for image identification (i.e. actions performed to multiple images at a time)	—	—	—	Yes
	Data filtering for image classification	—	Users can run a Python script to sort, move and organize images according to the MegaDetector predictions and a specified confidence threshold. Using their preferred tool, users can further classify images predicted to contain the 'Animal' category	—	By subproject, camera deployment, species and other search criteria provided by the user (e.g. image status or highlighted images)
	Automatic metadata import (e.g. date image was taken)	Per request	—	If specified by the user. It requires the Exiftool and implementation of functions in the MLWIC2 package	Yes

TABLE 1 (Continued)

Workflow criteria	Features	Conservation AI	MegaDetector	MLWIC2	Wildlife insights
Image data extraction	Extraction of image labels (assigned by human vision and AI) and image annotations	Image labels (by AI)	Image labels (by AI)	Image labels (by AI)	Image labels (by human vision and AI) and image annotations
Data exploration	Data summaries	Summaries of species records identified by Conservation AI	—	—	Summaries of species records, sites surveyed and sampling periods
	Data analysis	—	—	—	Features to provide detection rate, single species activity and activity overlap between species under development. Plans to include features to provide species richness, wildlife picture index and single- and multi-species occupancy estimates
Data export	Output format and output download	CSV, Excel or PDF file with rows for every image, folder names (image path), species name, confidence values (above a 0.5 threshold), sensor name and project name. Output shared with data owner after a request is sent to developers	MegaDetector provides a JSON file as output, which indicates the locations of detected objects in each image and associated confidence values for each detection	CSV file with rows for every image, folder names (image path), and the top five predictions for each image along with their associated confidence values	Four CSV files with data from the project, cameras, deployments and images. The images CSV file contains rows for every image, filename, metadata, species name and other fields recorded by the user. Data download is requested in Wildlife Insights and received via email
Documentation	Link	Conservation AI	MegaDetector	MLWIC2	Wildlife Insights
	References	Chalmers et al. (2019)	Beery et al. (2019)	Tabak et al. (2020)	Ahumada et al. (2020)

TABLE 2 Overview of the requirements, tools for data management, and data exploration associated with auxiliary platforms, Camelot and Timelapse, that integrate artificial intelligence (AI) output from MegaDetector for further camera-trap data processing

Workflow criteria	Features	Camelot	Timelapse
Overview	Description	Software for management and processing of camera-trap images. Camelot incorporates AI output from the MegaDetector model to facilitate camera-trap image processing	Software for management and processing of camera-trap images and videos. Timelapse incorporates AI output from the MegaDetector model to facilitate camera-trap image processing
	Developers	Camelot Team in consultation with Fauna & Flora International	Saul Greenberg at University of Calgary
Requirements	Cost	Free	Free
	Public data or shared with third party during analysis	No	No
	Internet access required	For integrating MegaDetector output into Camelot and for multi-user environments	Timelapse can be used offline once the software is downloaded and MegaDetector output is obtained
	Prerequisites/software	Java Runtime, Web browser (Chrome, Firefox, Edge), Camelot software	Timelapse software
	Operating system/ hardware supported	Specific releases for Windows, macOS, Linux and a Java .jar release that can be used with any operating system	Windows, Windows emulators and Windows virtual machines
	Data format	JPG, PNG	Images: JPG; Videos: AVI, MP4, ASF
	Memory/storage capacity	Minimum physical memory requirements of 2084 MB and 4096 MB for datasets of approximately 50,000 and 100,000 images	Unlimited. Based on local memory/ storage capacity
	Data upload/import options	Images imported to Camelot by browsing files on a local computer. Manual or CSV bulk import of camera deployment information, including geographical coordinates associated with each camera and the dates it was in operation	Images imported to Timelapse by browsing files on a local computer. Camera deployment information, including sites and dates of deployment is retrieved by Timelapse according to folder structure
AI module	How to use AI	Users must activate the 'wildlife detection' in Camelot to run MegaDetector. Users must provide an initial confidence threshold for assigning predictions made by computer vision. AI output can be used to filter images containing wildlife or people	Users must obtain the JSON file produced by MegaDetector and activate the 'automatic image recognition' option in Timelapse. AI output can be used to filter blanks, and images containing wildlife or people
	Interface for AI output validation	To review and annotate images, users can filter by confidence threshold, survey or trap	To review and annotate images, users can filter by confidence threshold, survey or trap
Data management	Data input schema	Users can specify a data-entry protocol and have complete control of any additional fields that they would like to record besides species classification	Users can specify a data-entry protocol and have complete control of any additional fields that they would like to record besides species classification
	Data management system (e.g. data organization by projects, deployments, locations)	Data can be organized by datasets, organization, surveys, sites, cameras. Images will be presented in the Camelot interface that serves as a dashboard to visualize and annotate images, selecting one or multiple images at a time	Data can be organized by datasets, surveys, sites and cameras. Images will be presented in the Timelapse interface that serves as a dashboard to visualize and annotate images, selecting one or multiple images at a time

TABLE 2 (Continued)

Workflow criteria	Features	Camelot	Timelapse
Image processing tools	Features for image editing (e.g. zooming, brightness and contrast) and image/metadata review	Image editing (brightness and contrast), and image metadata inspection	Image editing (brightness, contrast and sharpness), and image metadata inspection
	Multi-user environments	The project owner can give remote access to collaborators by sharing the 'Known URLs' of the project	Images can be split by regions, sites, etc., and different MegaDetector results files can then be generated for each group of images. The images and the MegaDetector results files need to be transferred to collaborators and stored locally where Timelapse will run. Alternatively, multi-user environments can be created using a virtual machine running Windows and Timelapse
	Bulk actions for image identification (i.e. actions performed to multiple images at a time)	Yes	Yes
	Data filtering for image classification	By survey, camera trap station, species name and other search criteria provided by the user (e.g. genus, common name)	By survey, camera trap station, species name and other search criteria provided by the user (e.g. date/time, image quality, count)
	Automatic metadata import (e.g. date image was taken)	Yes	Yes
Image data extraction	Extraction of image labels (assigned by human vision and AI) and image annotations	Image labels (by human vision and AI) and image annotations	Image labels (by human vision and AI) and image annotations
Data exploration	Data summaries	Summaries of species records, sites surveyed and sampling periods	—
	Assessment of independent records	Users can thin data using a specified temporal independence threshold and additional default rules in Camelot	No
	Data analysis	Provides a summary of the percentage of nocturnal images and a Relative Abundance Index. Generates summary tables that can be read into R using the CAMTRAPR package, or detection matrices required to fit occupancy models in Program PRESENCE	—
Data export	Output format and output download	CSV file with rows for every image, folder names (image path), metadata, default fields (species name, count, sex, life stage) and other fields specified by the user. The CSV output can be directly generated in Camelot	CSV file with rows for every image, relative path, folder names, metadata, species name and other fields specified by the user. The CSV output can be directly generated in Timelapse
Documentation	Link	Camelot	Timelapse
	References	Hendry and Mann (2018)	Greenberg et al. (2019)

Pros: Easily integrates with other platforms (Camelot, Timelapse, Zooniverse), which provide functionality for data organization/management, image review/annotation, multi-user environments and tools for validating AI predictions; provides different options for running the model depending on a user's computer and programming expertise; AI models for detection of animals, humans and vehicles are trained on a global scale; data sharing not required; not dependent on internet connection, can run locally.

Cons: No data repository available; does not normally provide species classifications or tools for model training by individual users.

3.2.1 | Timelapse

Timelapse is a software program for image processing that can be run offline, and in all versions of Microsoft Windows or other operating systems running Windows emulators. Timelapse incorporates AI results provided by MegaDetector to accelerate further data processing. Timelapse includes a *Template Editor* to allow the user to have complete control of any additional fields that they would like to record (e.g. vegetation characteristics associated with images or specific animal features). The Template Editor allows the user to specify a data-entry protocol, including the option of specifying data labels with default values and data-input controls that can prevent errors when multiple people are involved in processing the images (Greenberg et al., 2019). To divide work between collaborators, images can be split by regions, locations, etc., and different MegaDetector results files can then be generated for each group of images. The images and the MegaDetector results files can either be accessed directly via network drives or transferred to collaborators (e.g. using hard drives or file transfer utilities) and stored locally where Timelapse will run. Once images and the MegaDetector results are imported to Timelapse, users can start data processing and make use of the AI results to accelerate image revision. For example, users will be able to display all the images predicted as blanks by computer vision with high confidence, allowing these images to be easily selected and marked as blanks, and if desired, quickly reviewed for false negatives. Timelapse also provides an optional menu setting to classify images as nocturnal.

Pros: Data organization/management system, features for image review/annotation, multi-user environments, and tools for validating AI predictions; can incorporate MegaDetector output in data processing workflows; software stability; no internet connection or data sharing required.

Cons: No image repository or advanced reporting and analytical capabilities available; for multi-user environments, images need to be split and stored locally by each collaborator; only runs with the Windows operating system, Windows emulators, and Windows virtual machines.

3.2.2 | Camelot

Camelot was also developed for data management and processing purposes. It provides specific releases for Windows, macOS and Linux operating systems, and a Java .jar release can be used with any operating system. Users have to input camera deployment information, including a name and geographical coordinates associated with each camera and the dates it was in operation, either by providing the information via a Graphical User Interface or as a bulk CSV data import. Images can be imported to the software by browsing files on a local computer and will be presented in a *Library* that serves as a dashboard where the user can visualize images, select one or multiple images at a time, edit their brightness and contrast, and inspect metadata associated with each image. Users have complete flexibility when specifying data fields to be recorded when processing data.

Output from MegaDetector can be incorporated into Camelot to facilitate a semi-automated workflow, where users can filter images containing wildlife or people. This option requires a Camelot account and a good internet connection as it is an online service. After registering and uploading images to the cloud, users must activate the 'wildlife detection' option in the 'administration interface', and Camelot will automatically run MegaDetector on these images. When activating image recognition using MegaDetector, users must provide an initial confidence threshold for assigning predictions made by computer vision, but this threshold can be changed at any time.

Camelot includes an analytical module that provides a summary of the percentage of nocturnal images and a Relative Abundance Index. It also generates summary tables that can be read into R using the CAMTRAPR package for managing, visualizing and tabulating camera-trap data (Niedballa et al., 2016). Camelot can also output detection matrices that can be used to fit occupancy models in Program PRESENCE (Hines, 2006; MacKenzie et al., 2002), and it allows users to thin data using a specified temporal independence threshold (Iannarilli et al., 2019). Camelot's web interface allows multiple users to work on the same project; the project owner can give remote access to collaborators by sharing the 'Known URLs' displayed in the application. Camelot uses a Java virtual Machine to run and has minimum physical memory requirements of 2084 and 4096 MB for datasets of approximately 50,000 and 100,000 images. More details of memory limitations and options for working with large datasets can be found in the software documentation at https://camelot-project.readthedocs.io/en/latest/.

Pros: Data organization/management system, features for image review/annotation, multi-user environments, and tools for validating AI predictions; can incorporate MegaDetector output in data processing workflows; advanced reporting and analytical capabilities; internet connection is not required except when running AI models and when working with multiple collaborators; data sharing not required; works with most computer operating systems.

Cons: No image repository available; tasks (e.g. image upload, searching images and summarizing output) can slow down as the dataset increases in size; users might need to manually configure Java for more efficient memory allocation when running Camelot.

3.3 | MLWIC2: Machine learning for wildlife image classification

MLWIC2 is an R package developed for detecting and classifying species from North America ('species_model'), although is also useful for identifying blank images ('empty_animal' model) from different geographical regions (Tabak et al., 2020). MLWIC2 allows the user to run its AI models locally and to have an independent workflow without the need of image submission. Users need to install Anaconda Navigator, Python (3.5, 3.6 or 3.7), Rtools (for Windows computers) and version 1.14 of TensorFlow (Abadi et al., 2015) (see GitHub repository, https://github.com/mikeyEcology/MLWIC2). Users must know the R language and be familiar with file path specifications. MLWIC2 will provide an output file containing image filenames and the top five predictions for each image along with their associated confidence values. In addition, the R package provides functionality to train your own model using a subset of labelled images, which could be useful for improving AI performance. We illustrate the process used to train a model in the GitBook (Vélez & Fieberg, 2022) using a small set of images since training a model can be computationally intensive.

Pros: Provides a module for training your own model; has a Shiny App for interactively using its AI model, and training your own model; data sharing not required; not dependent on internet connection, can run locally.

Cons: Does not include tools for data organization/management, image review/annotation or multi-user environments; no image repository or advanced reporting and analytical capabilities available; requires more advanced computational skills and local computing power; the 'species_model' is geographically limited to species from North America.

3.4 | Conservation AI

Conservation AI is a cloud-based platform developed at the Liverpool John Moores University (UK) to help conservation projects use AI to process acoustic recordings, drone images, and camera-trap images and videos. It currently has trained models for identifying humans, man-made objects (e.g. cars and fires), and species from the United Kingdom, South Africa, North America and Tanzania. It provides services for image detection and classification in near-real time from linked devices capable of transferring images using a Simple Mail Transfer Protocol (SMTP). Any camera can be used for real-time detection as long as it supports SMTP and you have internet coverage in your study area. Alternatively, images can be directly uploaded to the platform using a batch upload of up to 1,000 images at a time. Once uploaded, images are classified using the available AI models, which can process approximately 10,000 images per hour. After running a particular model, images will be available in the platform with their corresponding AI prediction.

In addition to the currently available models, Conservation AI also provides a platform for image tagging and model training for specific datasets. Users can upload images directly into the tagging site or share them with the developers (e.g. via Google Drive) who will then upload batches of 500 images for you. For tagging, users will draw bounding boxes around animals in the images and label them with the species' name; this process will create the training dataset. Users will need to tag a minimum of 1,000 images per species, and the available models will be updated using transfer learning based on the new tags. The tagging section contains a species list with tags from different projects registered in the platform, and users can request to train models using any of the tagged data available in the platform. Conservation AI provides all of its functionality in the cloud, so a good internet connection is needed. This platform will output species identifications (for predictions above a 0.5 confidence threshold) along with associated confidence values for each record.

Pros: No programming expertise is required to run AI models; real-time detection capabilities; provides a module for training your own model and a multi-user environment for compiling the training dataset.

Cons: Does not include tools for data organization/management or image review/annotation; no data repository or advanced reporting and analytical capabilities available; users are dependent on developers' availability when training models; image upload is performed in small batches; trained models are geographically limited to species from United Kingdom, South Africa, North America and Tanzania; images uploaded to the platform will be accessible to Conservation AI registered users; cloud-based, which makes it susceptible to connection instability and service outages.

4 | MODEL EVALUATION

We evaluated the performance of AI platforms for animal detection (MegaDetector and the 'empty_animal' model of MLWIC2) and species classification (Conservation AI, 'species_model' from MLWIC2 and Wildlife Insights). We conducted out-of-sample validation by applying models to datasets not used in model training. These datasets included the Snapshot Kgalagadi and SWG Camera Traps 2018–2020 (SWG, 2021) datasets, both stored at the Labelled Information Library of Alexandria: Biology and Conservation (LILA-BC), and the Montana dataset collected by the Smithsonian's National Zoo and Conservation Biology Institute through an agreement with American Prairie (https://www.americanprairie.org/; no permit number). We also used our own data, collected in the Colombian Orinoquía (research permits issued to Universidad de los Andes, ANLA Resolution 1177, 2014; Cormacarena Resolution PM-GA. 3.20.2737) and archived in LILA-BC. Further details regarding the datasets can be found in Table 3, and for the LILA-BC datasets, in their corresponding repositories (https://lila.science/datasets).

Expert (i.e. human vision) labels were compared to classifications by the AI models associated with Wildlife Insights (predictions downloaded in July 2022), MegaDetector (MDv4.1), MLWIC2 (version 1.0) and Conservation AI (predictions downloaded in July 2022) to determine how these models perform when applied to data that were not included in their training datasets. The evaluation datasets contained very few images of humans or vehicles, and the images of humans were primarily associated with camera setup/take down. Therefore, we removed records containing 'human' or 'vehicle' classes. Workflows describing the use of the platforms, managing their output and comparing predictions with labels from classified images using R software v. 4.1.3 (R Core Team, 2021) are illustrated in an open-source GitBook (Vélez & Fieberg, 2022). Model performance was evaluated using functions in the CARET package

TABLE 3 Datasets used to evaluate artificial intelligence (AI) performance, with corresponding geographic region and number of images. Analysis level indicates whether classifications were assigned by experts at the image or sequence level. Species and animal classifier columns list the models assessed with each dataset. The Montana dataset was not run with MegaDetector or MLWIC2 'empty_animal' model as the dataset only contained images with animals and empty images were previously removed. Conservation AI was only run with snapshot Kgalagadi and the Montana dataset, as there were no other models available to classify species from Asia or South America

Dataset	Region	No. images	Analysis level	Species classifier	Animal classifier	Source
Montana	North America	5,122	Image	Wildlife Insights, MLWIC2, Conservation AI	—	https://www.americanprairie.org/
Orinoquía Camera Traps	South America	112,247	Image	Wildlife Insights, MLWIC2	MegaDetector, MLWIC2	https://lila.science/Orinoquia-camera-traps/
Snapshot Kgalagadi	Africa	10,222	Sequence	Wildlife Insights, MLWIC2, Conservation AI	MegaDetector, MLWIC2	https://lila.science/datasets/snapshot-kgalagadi
SWG Camera Traps 2018–2020	Asia	31,996	Sequence	Wildlife Insights, MLWIC2	MegaDetector, MLWIC2	https://lila.science/datasets/swg-camera-traps

TABLE 4 Metrics used to assess artificial intelligence (AI) model performance. True positives (TP): Number of observations where the species was correctly identified as being present in an image; true negatives (TN): Number of observations where the species was correctly identified as being absent in an image; false positives (FP): Number of observations where the species was absent, but the AI classified the species as being present; false negatives (FN): Number of observations where the species was present, but the AI classified the species as being absent

Metrics	Equation	Interpretation
Accuracy	(TP + TN)/(TP + FP + TN + FN)	Proportion of correct predictions in a dataset
Precision	TP/(TP + FP)	Probability the species is correctly classified as present given that the AI system classified it as present
Recall	TP/(TP + FN)	Probability the species is correctly classified as present given that the species truly is present
F1 Score	2 × precision × recall/(precision + recall)	Weighted average of precision and recall

(Kuhn, 2021) in R to estimate a confusion matrix for the observed and predicted classes as well as model precision, recall and F1 score (Table 4; Sokolova & Lapalme, 2009). We evaluated model performance at the taxonomic levels of species, genus, family, order and class. Model performance was estimated either at the image or the sequence level, depending on how the images were classified by experts (Table 3). We considered a range of confidence thresholds (0.1–0.99) when comparing models.

5 | RESULTS

The performance of MegaDetector and the MLWIC2 'empty_animal' model was dependent on the dataset, with the Orinoquía Camera Traps dataset having the highest F1 score (0.96 and 0.89 for MegaDetector and MLWIC2, respectively) and the Snapshot Kgalagadi dataset having the lowest F1 score (0.87 and 0.53 for MegaDetector and MLWIC2, respectively) when evaluated using a confidence threshold of 0.65 (Table 5). Precision and recall also varied by dataset and were generally lower for MLWIC2 than MegaDetector along a range of confidence thresholds between 0.1 and 0.99 (Figure 1); precision was particularly poor when applying MLWIC2 to the Kgalagadi dataset (Table 5). Recall was lowest for the SWG Camera Traps 2018-2020 dataset and highest for the Orinoquía Camera Traps dataset for both MegaDetector and MLWIC2 throughout the range of confidence thresholds evaluated (Figure 1).

We found few matches (i.e. labels shared) between the human and computer vision output for species classifiers (Conservation AI, MLWIC2 and Wildlife Insights), though additional matches could be identified at higher taxonomic levels, such as the family level (Figures 2 and 3; Table S1). Precision–recall curves for the different datasets showed that predictions for species classifications and higher taxonomic levels sometimes had high precision (>0.90). However, species classifications for nearly all of the datasets had low to moderate recall values (<0.70), suggesting that many of the individuals present in the images where missed (Figures 2 and 3).

Classifying images at higher taxonomic levels (e.g. at the genus, family, order or class level) typically increased the F1 score for the categories evaluated (reported, below, using a confidence threshold of 0.65). For Conservation AI, the highest F1 scores were obtained at the class level (91% and 95% for the Mammalia class in the Snapshot

Kgalagadi and Montana datasets, respectively) and at the order level (F1 score of 91% for the Artiodactyla order in the Snapshot Kgalagadi set) (Table S1). In addition to highly accurate predictions for the *Struthio camelus* species (F1 score of 99% in the Snapshot Kgalagadi set), Wildlife Insights also had relatively high F1 scores at the class (79% for Mammalia in the Orinoquía Camera Traps set), order (79% for Otidiformes in the Snapshot Kgalagadi set) and family (79% for Otididae in the Snapshot Kgalagadi set, 83% for Cuniculidae in the Orinoquía Camera Traps set) levels (Table S1). The highest F1 score for MLWIC2 was for the Mammalia class (77%).

6 | DISCUSSION

We found that common challenges associated with image recognition using AI, such as low accuracy when classifying species at new locations (Schneider, Greenberg, et al., 2020; Tabak et al., 2020), and variable model performance for different species (Whytock et al., 2021), were persistent even when using models trained with broad and diverse image datasets. Despite these challenges, AI-powered platforms that integrate AI output can help ecologists establish more efficient workflows for processing camera-trap images by providing tools for data management, image annotation, metadata extraction and data export (Greenberg, 2020). When evaluating AI platforms, users should consider model performance, platform requirements and built-in functionality (Table 1), as well as broader project needs. AI predictions can facilitate image processing by providing bounding boxes that help with animal localization, and accurate classifications, especially for broader categories such as the 'animal' category or higher taxonomic levels, could potentially be used to speed up processing by applying batch operations. For example, Fennell et al. (2022) found that using MegaDetector increased processing efficiency by 500% when compared to a fully manual workflow.

TABLE 5 Model performance metrics for the detection of animals in images using MegaDetector and the MLWIC2 'empty_animal' model when applied to the Snapshot Kgalagadi (KGA), Orinoquía camera traps (ORI) and SWG camera traps 2018–2020 (SWG) datasets. In each case, we used a confidence threshold of 0.65 when determining the classifications

Dataset	Animal classifier	Precision	Recall	F1
KGA	MegaDetector	0.82	0.93	0.87
KGA	MLWIC2	0.38	0.85	0.53
ORI	MegaDetector	0.98	0.93	0.96
ORI	MLWIC2	0.81	0.99	0.89
SWG	MegaDetector	0.99	0.78	0.87
SWG	MLWIC2	0.93	0.74	0.83

Predictions at higher taxonomic levels are an important contribution of AI platforms that can be leveraged by users to facilitate image processing in semi-automated workflows; these predictions could be used to subset and organize data for subsequent image review by humans, similar to subsetting images with an animal present using MegaDetector output or the 'empty_animal' model from MLWIC2. Using higher taxonomic levels could be particularly useful when the species of interest are not included in a species classifier, but the model is still capable of providing high accuracy at the family, order or class level (Tabak et al., 2022). For Conservation AI, the MLWIC2 'species_model' and Wildlife Insights, the highest F1 scores were found at the family, order and class level, likely due to the difficulty of discerning among closely related species with similar coloration or shape (Whytock et al., 2021).

When implementing a semi-automated workflow, it is important to consider that users can increase recall by selecting a lower confidence threshold for obtaining model predictions; decreasing the confidence threshold will reduce the proportion of animals (or species) missed by AI but at the expense of reducing precision (i.e. more false positives), which may require additional human involvement in image analysis. Projects incorporating AI in their workflows would benefit from examining a range of confidence thresholds similar

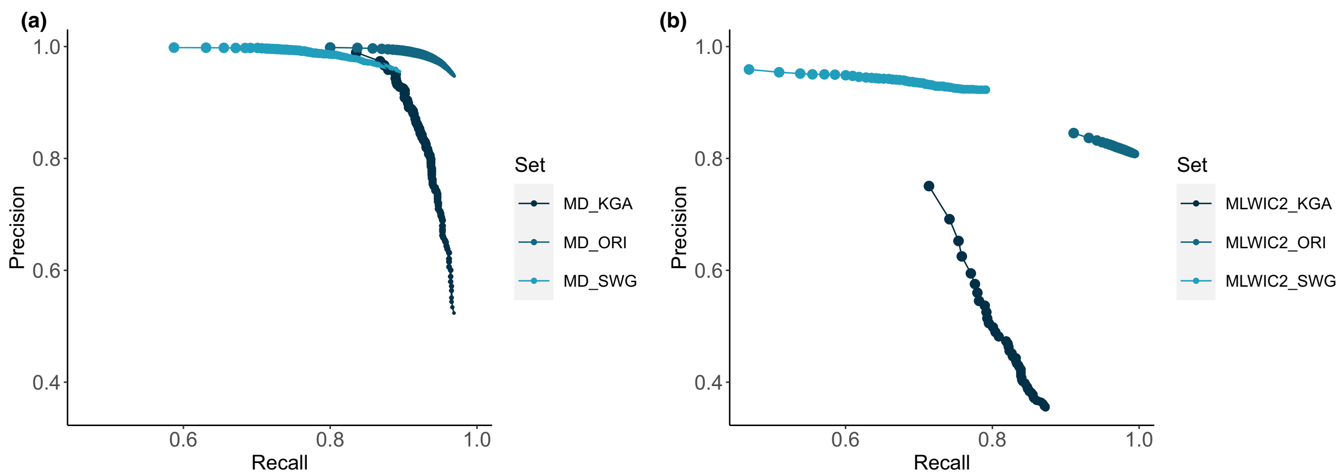

FIGURE 1 Precision and recall values for confidence thresholds (0.1–0.99 range) used to detect animals in images using MegaDetector (a) and MLWIC2 (b) when applied to the Snapshot Kgalagadi (KGA), Orinoquía Camera Traps (ORI) and SWG Camera Traps 2018–2020 (SWG) datasets. Larger points represent higher confidence thresholds.

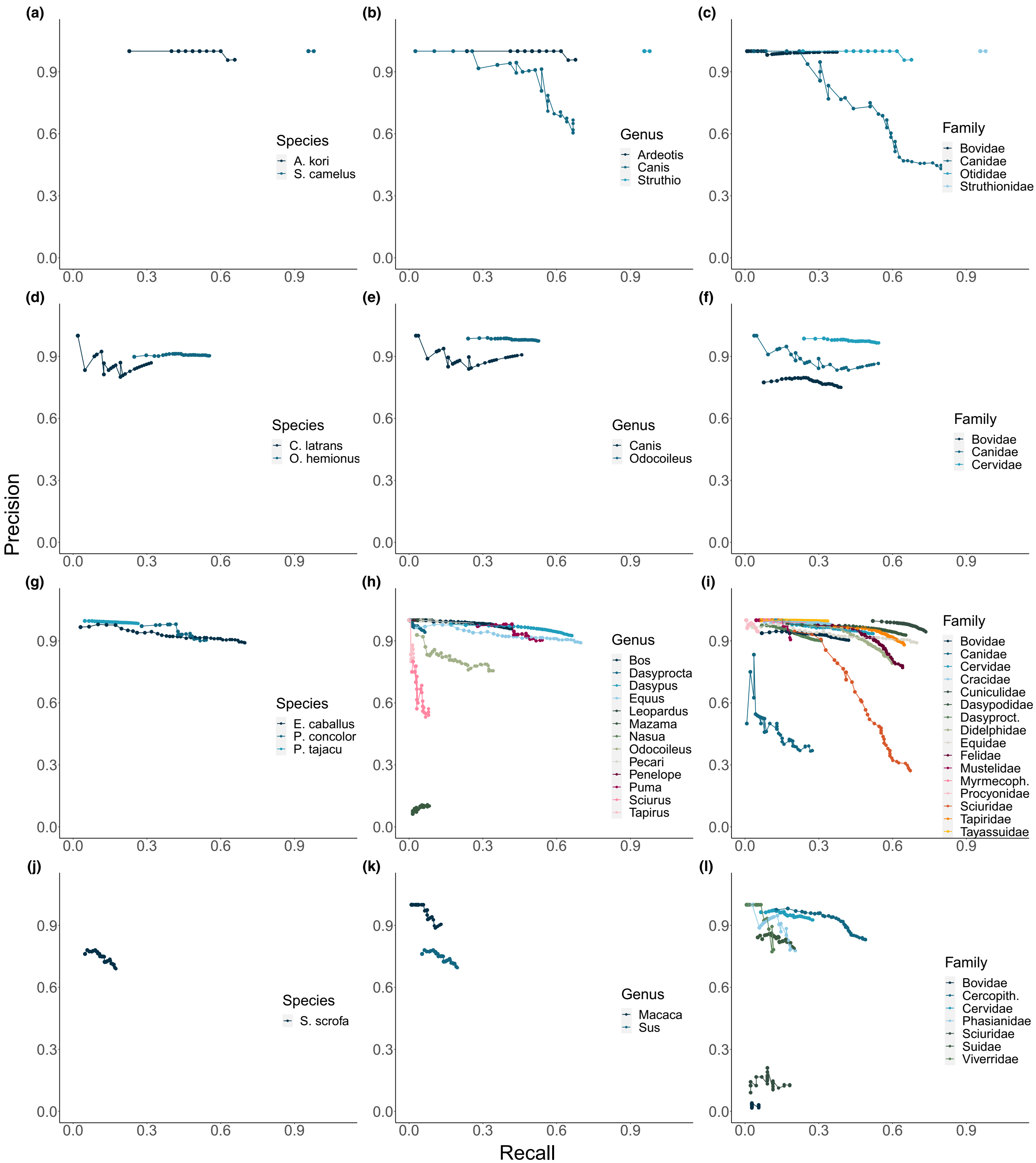

FIGURE 2 Precision and recall values for confidence thresholds (0.1–0.99 range) used to predict species, genus and family using Wildlife Insights. Larger points represent higher confidence thresholds. (a–c): Snapshot Kgalagadi, (d–f): Montana, (g–i): Orinoquía camera traps, (j–l): SWG camera traps 2018–2020. *Ardeotis kori* = *A. kori*, *Canis latrans* = *C. latrans*, Cercopithecidae = Cercopith., Dasyproctidae = Dasyproct., *Equus caballus* = *E. caballus*, Myrmecophagidae = Myrmecoph., *Odocoileus hemionus* = *O. hemionus*, *Pecari tajacu* = *P. Tajacu*, *Puma concolor* = *P. concolor*, *Struthio camelus* = *S. camelus*, *Sus scrofa* = *S. scrofa*.

to Figures 2 and 3 to determine how they impact precision and recall. For cases where recall values are low, experts would normally still want to review images to find the animals missed by computer vision. Users interested in developing a fully automated workflow for species classification will likely need to train their own models or retrain an existing model to improve model performance, for

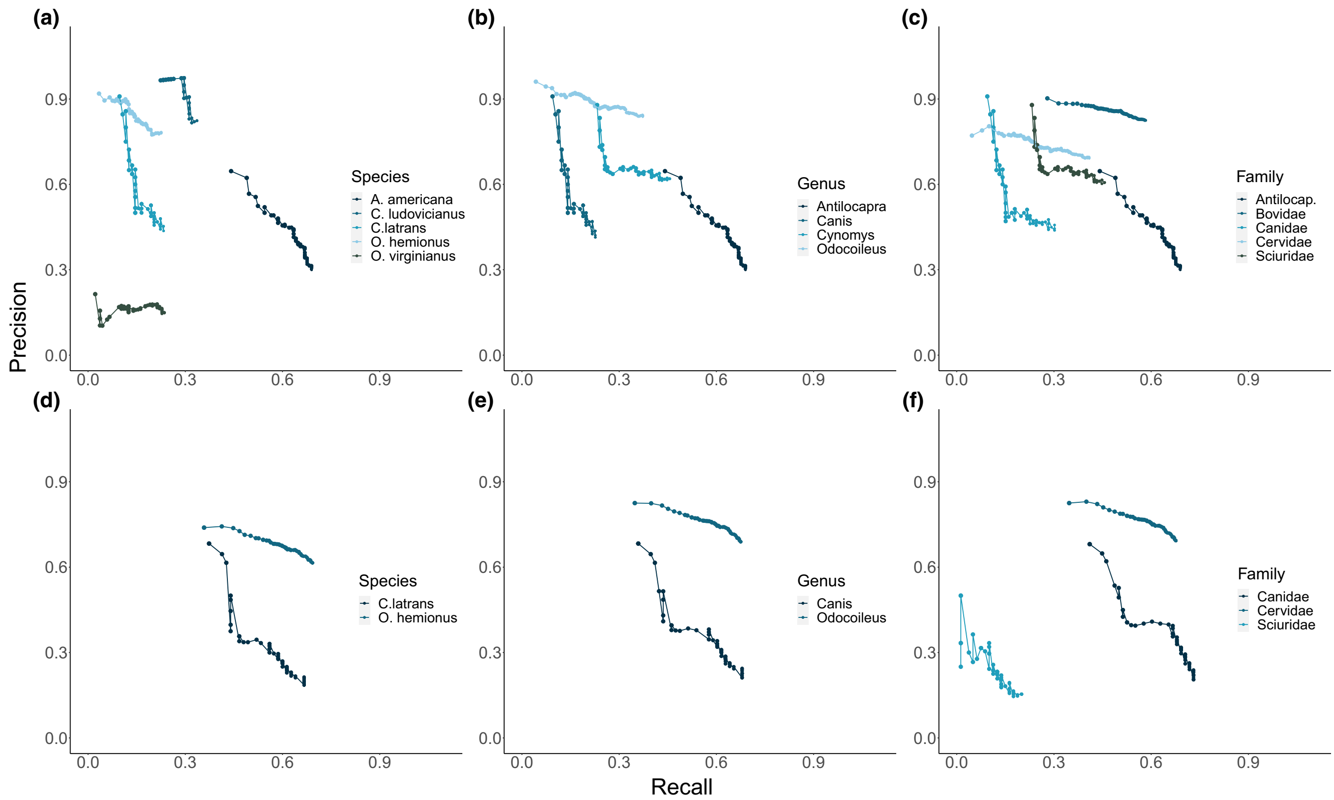

FIGURE 3 Precision and recall values for a range of confidence thresholds (0.1–0.99 for MLWIC2 'species_model' and 0.5–0.99 for Conservation AI) used to predict species, genus and family in the Montana set. Larger points represent higher confidence thresholds. (a–c): Model performance for MLWIC2, (d–f): Model performance for Conservation AI. Antilocapridae = Antilocap., *Antilocapra americana* = *A. americana*, *Canis latrans* = *C. latrans*, *Cynomys ludovicianus* = *C. ludovicianus*, *Odocoileus hemionus* = *O. hemionus*, *Odocoileus virginianus* = *O. virginianus*. Note, the minimum confidence threshold reported by Conservation AI is 0.5.

example, using the MLWIC2 package in R or Conservation AI's infrastructure. However, achieving high accuracy rates will still require access to a broad and reliable training dataset.

The development of AI models for species identification is an area of active research, and the platforms we have reviewed are undergoing continuous model development. AI models continue to be updated with new data and should lead to better model performance over time. For example, the newer version of MegaDetector (MDv5) increased processing speed, and incorporated additional training data to improve detection of the 'vehicle' class, artificial objects (e.g. bait stations), and particular taxa (rodents, reptiles and small birds). In addition to model updates, AI platforms continue to improve their features for storing and managing data, and for integrating AI output in data-processing workflows. These improvements will facilitate the review of AI classifications by users, allowing them to correct incorrect classifications, to add species labels to non-blank images and to capture other relevant information in the images (Greenberg, 2020; Whytock et al., 2021). To date, there has been little work to develop AI models that can identify individual characteristics (e.g. an animal's sex or age class) or behaviours (e.g. whether animals are feeding, moving or resting). We expect deep learning will also play a significant role in predicting these characteristics and behaviours once more data have been collected and made available for training new models.

AUTHOR CONTRIBUTIONS

Juliana Vélez and John Fieberg conceived the ideas and designed the methodology; Juliana Vélez, Hila Shamon and Paula J. Castiblanco-Camacho collected and processed the data; Juliana Vélez and John Fieberg analysed the data and led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

ACKNOWLEDGEMENTS

We thank Juan David Rodríguez and volunteers for their assistance with camera-trap data collection. We also appreciate the constant support from César Barrera and Eduardo Enciso who allowed us to set up camera traps on their properties and facilitated field expeditions. Thank you to American Prairie for allowing us to sample on their lands, and to John and Adrienne Mars for supporting Smithsonian's Great Plains Science Program. We acknowledge the valuable comments provided by Tanya Birch, Dan Morris, Saul Greenberg, and two anonymous reviewers, that considerably improved the manuscript. This research was made possible thanks to funding from the Colciencias-Fulbright Scholarship, the WWF's Russell E. Train Education for Nature Program (EFN), the Interdisciplinary Center for the Study of Global Change Fellowship, the Department of Fisheries, Wildlife and Conservation Biology at the University of Minnesota and the Smithsonian's National Zoo and Conservation Biology Institute. John Fieberg received partial salary support from the Minnesota Agricultural Experimental Station.

CONFLICT OF INTEREST

We have no conflicts of interest.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14044.

DATA AVAILABILITY STATEMENT

Camera-trap images and annotations are archived in the Labelled Information Library of Alexandria: Biology and Conservation (LILA-BC; https://lila.science/datasets). Code and guidelines to use AI platforms for processing camera-trap data are published in an open-source GitBook (Vélez & Fieberg, 2022).

ORCID

Juliana Vélez https://orcid.org/0000-0003-0412-2761
Michael A. Tabak https://orcid.org/0000-0002-2986-7885
John Fieberg https://orcid.org/0000-0002-3180-7021

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. https://www.tensorflow.org/

Ahumada, J. A., Fegraus, E., Birch, T., Flores, N., Kays, R., O'Brien, T. G., Palmer, J., Schuttler, S., Zhao, J. Y., Jetz, W., Kinnaird, M., Kulkarni, S., Lyet, A., Thau, D., Duong, M., Oliver, R., & Dancer, A. (2020). Wildlife insights: A platform to maximize the potential of camera trap and other passive sensor wildlife data for the planet. *Environmental Conservation*, *47*(1), 1–6. https://doi.org/10.1017/S0376892919000298

Augustine, B. C., Royle, J. A., Kelly, M. J., Satter, C. B., Alonso, R. S., Boydston, E. E., & Crooks, K. R. (2018). Spatial capture–recapture with partial identity: An application to camera traps. *The Annals of Applied Statistics*, *12*(1). https://doi.org/10.1214/17-AOAS1091

Beery, S., Liu, Y., Morris, D., Piavis, J., Kapoor, A., Joshi, N., Meister, M., & Perona, P. (2020). Synthetic examples improve generalization for rare classes. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 863–873.

Beery, S., Morris, D., & Yang, S. (2019). Efficient pipeline for camera trap image review. *arXiv Preprint arXiv:1907.06772*.

Beery, S., Van Horn, G., & Perona, P. (2018). Recognition in terra incognita. *Proceedings of the European Conference on Computer Vision (ECCV)*, 456–473.

Bogucki, R., Cygan, M., Khan, C. B., Klimek, M., Milczek, J. K., & Mucha, M. (2018). Applying deep learning to right whale photo identification. *Conservation Biology*, *33*(3), 676–684. https://doi.org/10.1111/cobi.13226

Burton, A. C., Neilson, E., Moreira, D., Ladle, A., Steenweg, R., Fisher, J. T., Bayne, E., & Boutin, S. (2015). REVIEW: Wildlife camera trapping: A review and recommendations for linking surveys to ecological processes. *Journal of Applied Ecology*, *52*(3), 675–685. https://doi.org/10.1111/1365-2664.12432

Carl, C., Schönfeld, F., Profft, I., Klamm, A., & Landgraf, D. (2020). Automated detection of European wild mammal species in camera trap images with an existing and pre-trained computer vision model. *European Journal of Wildlife Research*, *66*(4), 1–7. https://doi.org/10.1007/s10344-020-01404-y

Chalmers, C., Fergus, P., Wich, S., & Montanez, A. C. (2019). Conservation AI: Live stream analysis for the detection of endangered species using convolutional neural networks and drone technology. *arXiv Preprint arXiv:1910.07360*.

Chen, P., Swarup, P., Matkowski, W. M., Kong, A. W. K., Han, S., Zhang, Z., & Rong, H. (2020). A study on giant panda recognition based on images of a large proportion of captive pandas. *Ecology and Evolution*, *10*(7), 3561–3573. https://doi.org/10.1002/ece3.6152

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

Christin, S., Hervet, É., & Lecomte, N. (2019). Applications for deep learning in ecology. *Methods in Ecology and Evolution*, *10*(10), 1632–1644. https://doi.org/10.1111/2041-210x.13256

Efford, M. G., & Fewster, R. M. (2013). Estimating population size by spatially explicit capture–recapture. *Oikos*, *122*(6), 918–928. https://doi.org/10.1111/j.1600-0706.2012.20440.x

Fennell, M., Beirne, C., & Burton, A. C. (2022). Use of object detection in camera trap image identification: Assessing a method to rapidly and accurately classify human and animal detections for research and application in recreation ecology. *Global Ecology and Conservation*, *35*, e02104. https://doi.org/10.1016/j.gecco.2022.e02104

Gomez, A., Diez, G., Salazar, A., & Diaz, A. (2016). Animal identification in low quality camera-trap images using very deep convolutional neural networks and confidence thresholds. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, F. Porikli, S. Skaff, A. Entezari, J. Min, D. Iwai, A. Sadagic, C. Scheidegger, & T. Isenberg (Eds.), *Advances in visual computing* (pp. 747–756). Springer International Publishing.

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

Greenberg, S. (2020). *Automated image recognition for wildlife camera traps: Making it work for you* [Technical report]. University of Calgary. http://hdl.handle.net/1880/112416

Greenberg, S., Godin, T., & Whittington, J. (2019). Design patterns for wildlife-related camera trap image analysis. *Ecology and Evolution*, *9*(24), 13706–13730. https://doi.org/10.1002/ece3.5767

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. (2017). On calibration of modern neural networks. *arXiv Preprint arXiv:1706.04599*.

Hendry, H., & Mann, C. (2018). Camelot—Intuitive software for camera-trap data management. *Oryx*, *52*(1), 15. https://doi.org/10.1017/S0030605317001818

Hines, J. E. (2006). *PRESENCE2, software to estimate patch occupancy and related parameters*. [Computer Program]. USGS – PWRC http://www.mbr-pwrc.usgs.gov/software/presence.html

Iannarilli, F., Arnold, T. W., Erb, J., & Fieberg, J. R. (2019). Using lorelograms to measure and model correlation in binary data: Applications to ecological studies. *Methods in Ecology and Evolution*, *10*(12), 2153–2162.

Kuhn, M. (2021). *Caret: Classification and regression training*. https://CRAN.R-project.org/package=caret

MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Andrew Royle, J., & Langtimm, C. A. (2002). Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, *83*(8), 2248–2255.

McShea, W. J., Forrester, T., Costello, R., He, Z., & Kays, R. (2016). Volunteer-run cameras as distributed sensors for macrosystem mammal research. *Landscape Ecology*, *31*(1), 55–66. https://doi.org/10.1007/s10980-015-0262-9

Niedballa, J., Sollmann, R., Courtiol, A., & Wilting, A. (2016). camtrapR: An r package for efficient camera trap data management. *Methods in Ecology and Evolution*, *7*(12), 1457–1462. https://doi.org/10.1111/2041-210X.12600
Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(25), E5716–E5725.
R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/
Royle, J. A., Chandler, R. B., Sollmann, R., & Gardner, B. (2013). *Spatial capture-recapture*. Academic Press.
Schneider, S., Greenberg, S., Taylor, G. W., & Kremer, S. C. (2020). Three critical factors affecting automated image species recognition performance for camera traps. *Ecology and Evolution*, *10*(7), 3503–3517. https://doi.org/10.1002/ece3.6147
Schneider, S., Taylor, G. W., & Kremer, S. (2018). Deep learning object detection methods for ecological camera trap data. *2018* 15th conference on computer and robot vision (CRV), 321–328.
Schneider, S., Taylor, G. W., & Kremer, S. C. (2020). Similarity learning networks for animal individual re-identification-beyond the capabilities of a human observer. *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, 44–52.
Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002
SWG. (2021). Northern and central annamites camera traps 2.0. *IUCN SSC Asian wild cattle specialist Group's Saola working group. Dataset*.
Tabak, M. A., Falbel, D., Hamzeh, T., Brook, R. K., Goolsby, J. A., Zoromski, L. D., Boughton, R. K., Snow, N. P., VerCauteren, K. C., & Miller, R. S. (2022). Camera- TrapDetectoR: Automatically detect, classify, and count animals in camera trap images using artificial intelligence. *bioRxiv*. https://doi.org/10.1101/2022.02.07.479461
Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Newton, E. J., Boughton, R. K., Ivan, J. S., Odell, E. A., Newkirk, E. S., Conrey, R. Y., Stenglein, J., Iannarilli, F., Erb, J., Brook, R. K., Davis, A. J., Lewis, J., Walsh, D. P., Beasley, J. C., VerCauteren, K. C., Clune, J., & Miller, R. S. (2020). Improving the accessibility and transferability of machine learning algorithms for identification of animals in camera trap images: MLWIC2. *Ecology and Evolution*, *10*(19), 10374–10383. https://doi.org/10.1002/ece3.6692
Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Sweeney, S. J., Vercauteren, K. C., Snow, N. P., Halseth, J. M., Di Salvo, P. A., Lewis, J. S., White, M. D., Teton, B., Beasley, J. C., Schlichting, P. E., Boughton, R. K., Wight, B., Newkirk, E. S., Ivan, J. S., Odell, E. A., Brook, R. K., … Photopoulou, T. (2018). Machine learning to classify animal species in camera trap images: Applications in ecology. *Methods in Ecology and Evolution*, *10*(4), 585–590. https://doi.org/10.1111/2041-210x.13120
Vélez, J., & Fieberg, J. (2022). *Guide for using artificial intelligence systems for camera trap data processing*. https://ai-camtraps.netlify.app/
Whytock, R. C., Świeżewski, J., Zwerts, J. A., Bara-Słupski, T., Koumba Pambo, A. F., Rogala, M., Bahaa-el-din, L., Boekee, K., Brittain, S., Cardoso, A. W., Henschel, P., Lehmann, D., Momboua, B., Kiebou Opepa, C., Orbell, C., Pitman, R. T., Robinson, H. S., & Abernethy, K. A. (2021). Robust ecological analysis of camera trap data labelled by a machine learning model. *Methods in Ecology and Evolution*, *12*(6), 1080–1092. https://doi.org/10.1111/2041-210X.13576
Young, S., Rode-Margono, J., & Amin, R. (2018). Software to facilitate and streamline camera trap data management: A review. *Ecology and Evolution*, *8*(19), 9947–9957. https://doi.org/10.1002/ece3.4464

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Vélez, J., McShea, W., Shamon, H., Castiblanco-Camacho, P. J., Tabak, M. A., Chalmers, C., Fergus, P., & Fieberg, J. (2023). An evaluation of platforms for processing camera-trap data using artificial intelligence. *Methods in Ecology and Evolution*, *14*, 459–477. https://doi.org/10.1111/2041-210X.14044