UC Agriculture & Natural Resources
Proceedings of the Vertebrate Pest Conference

## Title
Using Camera Traps and AI to Improve Efficacy and Reduce Bycatch at Goodnature A24 Rodent Traps in Hawaii

## Permalink
https://escholarship.org/uc/item/1k04f07v

## Journal
Proceedings of the Vertebrate Pest Conference, 29(29)

## ISSN
0507-6773

## Authors
Crampton, Lisa H.
Gallerani, Erica M.
Reeves, Mari K.

## Publication Date
2020

eScholarship.org Powered by the California Digital Library
University of California

# Using Camera Traps and AI to Improve Efficacy and Reduce Bycatch at Goodnature A24 Rodent Traps in Hawaii

**Lisa H. Crampton** and **Erica M. Gallerani**
Kauai Forest Bird Recovery Project, Pacific Cooperative Studies Unit, Hanapepe, Hawaii
**Mari K. Reeves**
Pacific Islands Fish and Wildlife Office, U.S. Fish and Wildlife Service, Honolulu, Hawaii

**ABSTRACT:** Camera traps provide an unobtrusive means to monitor wildlife presence and behavior. Yet there is a steep learning curve associated with their deployment. Camera model, settings and position, target behavior, and technicians' skill greatly influence the success of camera trapping. Furthermore, data storage and management are complex, as copious photos occupy considerable storage space. Finally, evaluating large numbers of digital images is time-consuming for low frequency events; in each of our trials we amassed 10,000-50,000 photos, of which 6-20% were target animals. The application of artificial intelligence (AI) to digital image datasets can greatly increase efficiency, but few existing algorithms have been trained on small animals. We embarked on a camera trapping project to assess interactions of target (rodent) and non-target (bird) species with 125 GoodNature A24 rat traps deployed in rainforest sites on Kauai, Hawaii, following several observations of non-target mortality. While our long-term goal was to use camera trap data to suggest modifications to traps that would maintain target kills while minimizing bycatch, the short-term goal presented in this manuscript focused on perfecting our camera trapping program and AI to classify photos of small animals. Specifically, we described lessons learned regarding 1) the performance of several camera models, 2) camera placement, 3) data management, and 4) artificial network training and development. First, we report on field studies assessing Bushnell TrophyCam HD, Bushnell HD, Reconyx HyperFire, and Reconyx HyperFire2 models on a variety of settings, distances, and angles with respect to the traps. Camera model and placement at traps are critical to capturing images amenable to AI development, as is variation in the training dataset. Second, we outline our data management and sharing protocols. Third, we describe the development of preliminary AI models to review and sort camera trap data. Early models reduced the workload of reviewing camera trap data by correctly classifying photos of rats, birds, humans, pigs, and empty frames. We expect these results to further improve with more training data. These results will greatly enhance the efficacy of several camera trapping studies that we have recently undertaken and help us modify traps to avoid bycatch.

**KEY WORDS:** A24 trap, algorithm, birds, Bushnell, camera, convoluted neural network, Kauai, photo sorting, rats, *Rattus*, Reconyx

**Proceedings, 29th Vertebrate Pest Conference** (D. M. Woods, Ed.)
Paper No. 53. Published December 10, 2020. 7 pp.

## INTRODUCTION

Invasive species, such as rats, are one of the main drivers of island endemic extinction, and the Hawaiian archipelago is not immune to their effects; 63% of Hawaii's endemic birds have gone extinct (Leonard 2008). On the island of Kauai, five of 13 historically known songbirds have disappeared in the last 40 years. For the eight remaining songbird species, invasive rats (*Rattus* spp.) have been implicated in reduced female, juvenile, and nestling survival (VanderWerf et al. 2014, Hammond et al. 2015, Hammond et al. 2016, Paxton et al. 2018). The endangered Puaiohi (*Myadestes palmeri*), which is endemic to Kauai, numbers 487 (95% CI: 405-579) individuals and is geographically restricted (Crampton et al. 2017). Rodent predation is likely the primary factor limiting this species (USFWS 2006, Atkinson et al. 2014, VanderWerf et al. 2014, Paxton et al. 2018). The Puaiohi experiences poor female and juvenile survival due to rat predation on nests and fledglings (Tweed et al. 2006, VanderWerf et al. 2014). Rats and mice also compete with forest birds by consuming fruits and invertebrates. Thus, effective rodent control to benefit Puaiohi and other songbirds is a high priority for the state of Hawaii (Hawaii DLNR 2015).

Since 2015, we have suppressed rodent populations with Goodnature A24 rat traps in critical bird habitat in rainforest sites on the Alakai Plateau on Kauai. Approximately 325 traps are active at two grids in the Alakai. Monitoring with ink plates has shown a four-fold decrease in relative rodent abundance on trapping grids versus reference plots (L. H. Crampton, unpubl. data) over this time period but we have not had the resources to investigate whether this level of suppression has led to significant increases in bird survival, and we continue to observe signs of predation on trapping grids (L. H. Crampton, unpubl. data). Meanwhile, we recently discovered that several birds, including Puaiohi, had been killed by A24 traps, which raised the question of whether the benefits of rodent control in terms of bird survival outweighed any inadvertent mortality or injury of birds or other non-target species.

To answer this question, we urgently needed to determine a) how often non-target species were hurt or killed by A24 traps, and b) how to modify traps to avoid bycatch while maximizing rodent control. To do so, we implemented several studies in which we modified traps by raising them or adding blockers and manipulated lure type to reduce attractiveness to birds. To help us interpret the results of these studies, we decided to deploy wildlife camera traps at individual rodent traps. Assessing which cameras to deploy and what settings and placement to use to record small animals in tropical environments, and how to manage and process camera trap data from remote areas, became a project of its own, the results of which are described here.

Camera traps have tremendous promise as an unobtrusive means to monitor wildlife presence and behavior (e.g., Rovero et al. 2013, Trolliet et al. 2014, Newey et al. 2015). Yet there is a learning curve associated with their deployment. Moreover, they generate enormous amounts of photo data that must be managed, sorted, and stored (Newey et al. 2015). Thus, we turned to artificial intelligence (AI) to increase our efficiency in identifying photos containing animals (e.g., rodents, birds) of interest. Because most efforts in this field have focused on large animals (e.g., Tabak et al. 2018, Falzon et al. 2020, Weideng et al. 2020) we needed to amass a training dataset for small animals in a tropical environment to train new neural network algorithms, which required us to manage, share, and manually sort tens of thousands of photos. The purpose of this paper is to describe what we learned in terms of a) the performance of various camera models, b) proper placement of cameras on traps, c) management of photo data, and d) development of AI to speed up photo processing. We hope this effort will help other researchers better implement camera trapping and AI to improve their trapping and pest control efforts. In a sense, it is an investigation of "best management practices" for camera trapping for small animals in tropical rain forests. The results of the trap modifications informed by this project will be presented in a second manuscript currently in preparation.

## METHODS AND RESULTS

### Study Area

The project focuses on the eastern portion of the Alakai Plateau on Kauai, Hawaii (22°7'18"N and 159°33'48"W), a ~70km$^2$ area of relatively pristine, wet (> 6,000 mm rain per year) montane forest dominated by ohia lehua (*Metrosideros polymorpha)*. Other tree and shrub species include olapa (*Cheirodendron trigynum*), lapalapa (*C. platyphyllum*), ohia ha (*Syzygium sandwicensis*), kawau (*Ilex anomala*), ohelo (*Vaccinium calycinum*), and kanawao (*Broussaisia arguta*). Many of these species bear flowers and fruit important to native birds as food but also consumed by rats. This area, which is remote, roadless and drained by numerous deeply incised streams, is owned and managed as a wilderness area and forest reserve by the State of Hawaii. The camera trapping studies were conducted at a 125-trap grid covering ~80 ha of this forest where all eight remaining songbird species native to Kauai occur.

### Camera Trapping

We conducted several different studies in the winter of 2018-2019 to assess the impact of several variables on camera performance: camera model, distance from target, horizontal offset, video vs. still, and sensitivity. These issues have been reviewed by other authors (e.g., Rovero et al. 2013, Trolliet et al. 2014, Newey et al. 2015), but their focus has been on large animals. One question we attempted to answer was whether small animals, such as birds (~15-40 g) and rats (~70-300 g) would trigger the motion sensor in a tropical environment (i.e., would the difference between their body heat and the background environmental temperature be great enough, or would we need to use the time lapse setting set at a frequent trigger interval?). In all studies, the lens of the camera was approximately level with the bottom of the trap (i.e., approximately 12 cm high; it was not above the trap or vertically angled down at the trap). In future studies, we will experiment with angled traps to reduce the background noise from plants and glare from the flash. Following the suggestions of Newey et al. (2015), we programmed all cameras in the office prior to deployment in the field, then double checked the settings in the field. We used lithium batteries in all cameras, as recommended by the user manuals.

In the first study, we compared the performance of three Bushnell Trophy Cam HDs and three Reconyx Hyperfire PC800s. Both cameras are equipped with a motion sensor mode, in which the interruption of an infrared beam by movement of a warm object in the camera frame (e.g., an animal or a sunlit leaf moving) causes the camera to take a picture; they can also be programmed to take photos at set intervals (time lapse mode). Performance was measured in terms of motion sensor efficacy (number of photos taken in motion sensor mode) and durability (total number of photos taken). Cameras were deployed in pairs (one of each brand) on traps, with the settings described in Table 1 and Table 2. In some cases, a third camera was set on video to assess how much action was missed by the still photos. Over several weeks, we experimented with different distances between the camera and the trap (1-3 m). In the second study, we experimented with the time lapse setting (1 min vs. 60 min) with five pairs of two Reconyx cameras pointed at the same object. The goal of this experiment was to determine if the motion sensor was working well enough to allow us to rely on it instead of the 1-min time lapse, which generated thousands of photos and rapidly depleted the batteries (similar to Newey et al. 2015).

These two studies showed that the Bushnell TrophyCams did not capture as many photos on motion sensor mode as the Reconyx Hyperfires. Also, Bushnell

**Table 1. Bushnell TrophyCam HD settings in December 2018 - January 2019.**

| Parameter | Setting |
|---|---|
| MODE | Camera |
| Image size | 5M |
| Image format | Full Screen |
| Capture Number | 3 photo |
| LED control | Medium |
| Camera name | N/A |
| Video size | 640X480 |
| Video length | 10S |
| Interval | 10S |
| Sensor Level | High |
| Format | No* |
| TV Out | NTSC |
| Time Stamp | On |
| Set Clock | Change if necessary |
| Field Scan | On |
| A | Start: 0:00 Stop: 23:59 |
| Interval | 1 min |
| Coordinate Input | Off |
| Video Sound | Off |
| Default Set | Cancel |

**Table 2. Reconyx Hyperfire settings in December 2018 - January 2019; the only change thereafter was to use a 60-min time lapse.**

| Category | Parameter | Settings |
|---|---|---|
| Trigger | Motion Sensor | On |
| Trigger | Sensitivity | High |
| Trigger | Pics per Trigger | 3 |
| Trigger | Picture Interval | RapidFire |
| Trigger | Quiet Period | No |
| Time Lapse | AM period | On /Start: 12:00 am Stop: 12:00 pm |
| Time Lapse | PM period | On/Start: 12:00 pm Stop: 12:00 am |
| Time Lapse | Interval | 1 min |
| Resolution | Resolution | 3.1MP |
| Night Mode | Night Mode | Balanced |
| Night Mode | Illuminator | On |
| Date/Time/Temp | Y/Mon/D/H/Min/Temp | Correct date and time and F |
| Codeloc | Codeloc | Ignore |
| User Label | User Label | View and change if doesn't match |
| Use Defaults | Use Defaults | Cancel |

cameras often stopped taking pictures shortly after they were set. Therefore, we discontinued use of the Bushnells, although they were more affordable. For our purposes, we preferred to spend more money on fewer, more reliable cameras (see Newey et al. 2015 for discussion of Reconyx vs. Bushnell). Furthermore, a high-quality image such as produced by the Reconyx was a paramount concern for future image recognition algorithms (Rovero et al. 2013).

They also showed that the Reconyx motion sensors captured most of the activity at traps (mean 227 photos, range 40-520 photos on motion sensor vs. mean 26 photos, range 10-60 photos on time lapse). In subsequent studies, we stopped using the 1-min time lapse in favor of a 60-min time lapse setting that acted as a "time stamp" so we would know when the battery failed. On the 1-min time lapse, the lithium batteries lasted less than two weeks. On the 60-min time lapse, they lasted greater than 16 weeks (they were still working when we removed the cameras). The reliability that we observed from the Reconyx motion sensors contradicts the results of Newey et al. (2015), who found that the Bushnells they deployed in motion sensor mode failed to detect 49-68% of the animals documented by paired cameras set in time-lapse mode. This discrepancy may reflect the professional quality of the Reconyx cameras.

We determined that a 1-m distance between camera and trap was optimal. The camera was close enough to be triggered by the animal of interest and the photo had little background "noise" but was not so close that the animal was out of focus or obscured by the reflection of the flash. We also learned that cameras should face the trap along a north-south axis to not be blinded by sun at its lowest angles (Newey et al. 2015). It is critical to thoroughly clear vegetation in front of the camera and behind the trap so that it does not falsely trigger the camera in response to photosynthesis or cause reflected sunlight to obscure the camera lens. It is better to attach the camera to a tree or rectangular stake than a smooth, round post such as PVC, because round posts allow the camera to move and lose focus on the object of interest (in this case the rodent trap).

Finally, it is essential to ensure that the rodent trap is in the center of the camera frame. The best option is to use the "walk test" function of the Reconyx camera; after setting the camera in this mode, the technician should move to the trap, cause some motion around the trap, then remove the data storage card from the camera and put it in a card viewer or second camera (if weather permits). Alternatively, if the weather is poor and one does not want to open up cameras, one can put a second camera (e.g., a point in shoot or even the camera of a mobile phone) in front of the trail camera, take a picture, and review the photo on the second camera's display. The time spent ensuring proper position during camera trap deployment far outweighs the cost of realizing when cameras are retrieved that they were focused on the wrong object for several weeks.

In our last study, we deployed one Reconyx Hyperfire (HF) and two Reconyx Hyperfire 2s (HF2) at each of eight randomly selected A24 traps. The goal was to investigate the effect of camera model, sensitivity, and position with respect to the trap on camera performance. We placed all cameras 1 m from the rodent trap, pairing one HF and one HF2 directly in front of the trap and next to each other (to compare performance of HFs and HF2s), and one HF2 at an offset or oblique angle to the rodent trap, 14-45 cm away from the paired cameras (to assess impact of environmental variability in the camera frame on the number of photos captured). For the first five days, one of the HF2s was set on "high" sensitivity and the other on "very high"; the HF was left at its highest setting, "high." We randomly alternated the different HF2 sensitivity settings between the two positions (centered vs. offset). After five days, all cameras were switched to the "high" setting. We used three different mixed effects regressions with these variables (camera model, sensitivity setting, and offset position) as fixed effects and trap location as a random effect; number of photos was the dependent variable.

This analysis showed that the HF2s performed better than the HFs, taking on average 284 ±1.15 photos (212-380) vs. 361 ±1.15 (270-482) photos per deployment ($b = 0.24 \pm 0.02$, $p < 0.001$), with the very high sensitivity setting capturing more photos than the high setting on the HF2s: 351 ±1.16 (219-392) photos vs. 293 ±1.16 (260-469) photos per deployment ($b = -0.18 \pm 0.03$, $p < 0.001$). The offset position resulted in fewer photos than the centered positions (284 photos vs. 340 photos on average; $p < 0.001$). Consequently, we determined that it was not possible to mix and match Reconyx models and settings in future camera trapping projects, where we wanted to use the number of photos as an index of animal activity around the trap, and that it was best to position the cameras in front of the trap (Trolliet et al. 2015).

A final lesson learned was that there was a lot of variation among traps in the amount of animal activity. Thus, it was important to try to place cameras at every trap, or at least most traps, and stratify the traps by environmental variables that might influence animal activity. If there are not enough cameras available to cover most or all traps simultaneously, we recommend moving cameras to new traps every few weeks. In our study, we divided our traps into five groups. The 16 HF2s were randomly assigned to 16 of the traps within the group. Approximately every 14 days, we moved the 16 cameras to a new group, such that over three months, cameras were placed on traps in all five groups. This practice increased our sample size of traps sampled, so it improved our power to detect differences among traps, even though the sample size in terms of total number of photos captured may have been similar to an experimental design in which cameras were left at the same trap for three months. It was also important for the neural network models to experience variation due to their tendency to overfit specific photo data, as described below.

**Data Management**

Camera trapping generates large amounts of data, stored on easily lost or misattributed data storage (SD) cards. A plan must be developed to track locations of cameras and the SD cards associated with each deployment for each camera. We assigned an ID name and two SD cards, A and B, to each camera (for example, camera 001 had cards labeled 001A and 001B). We also used SD card wallets for transportation, organization, and storage of cards. Before beginning field work, we entered all cameras by ID name into a cloud-based database developed by Natural Resource Data Solutions (NRDS; https://nrdsdata.com/). We also recommend using the "User Label" feature in Reconyx cameras to label each photo with the camera ID. In the field, we tracked the deployment location of those cameras using a spatially enabled mobile application provided by NRDS. In the application, we noted camera ID, GPS location, card ID, and variables such as camera offset, time lapse, sensitivity, battery level, and number of photos taken. These data were uploaded to the NRDS database.

Upon first deployment, the camera's "A" SD card was used, when it was time to check or move the camera, the "A" card was removed and replaced with its "B" card. The A card was taken back to the office, downloaded, reformatted, and readied for redeployment on the third camera check. Downloaded photos were categorized in Windows Explorer folders by date, trap ID and camera ID. Wallets always contained two-three extra cards in case some SD cards became corrupted and could not be read by the camera.

The size of the dataset was formidable at 24 GB with hundreds of thousands of photos taken by the 16 cameras deployed during the six-month study. The raw data from field cameras typically contained large numbers of false positives or motion-triggered photos that were not of interest (e.g., non-target animals or humans). For example, in one dataset of 56,500 photos collected with Reconyx HFs set on motion sensor +60-min time lapse setting, only 3,500 (6.2%) contained animals of interest; the remainder were mostly empty time lapse photos, with a few consisting of plants moving and people walking by. However, the 16,095-photo dataset from the HF vs. HF2 comparison (also on motion sensor +60-min time lapse setting) consisted of 20% (3,279 photos) of rats, cats, mice, and pigs, perhaps because of the greater sensitivity of the HF2s. The 1-min time-lapse we used in early deployments to help gauge sampling time until battery failure also increased the number of empty frames. Empty frames can cause a substantial drain on resources including storage capacity and time required for image sorting (Newey et al. 2015). Each field technician underwent a learning curve to proficiently deploy cameras to limit the number of unwanted photos of moving vegetation or photos not focused on traps. The transition to hourly time-lapse was a big time-saver.

To develop training data for neural network development we sorted photos manually. From January-September 2019, we sorted photos in Windows Explorer folders, using a variety of software available on Windows. We tracked progress in an Excel spreadsheet containing camera deployment data we exported from the NRDS database. For each camera deployment folder, we counted the number of photos in the folder and entered that in the corresponding camera deployment row in the spreadsheet. If there were animals in the photo, we recorded the species of animal in the photo, the number of pictures, the photo ID of the first photo with an animal, the photo ID of the last photo with an animal, whether the animal tripped the motion sensor, how many photos were in the animal series, the time in 24:00 format, the temperature, and whether the camera was on its day or night setting. We then moved photos with animals into new folders: birds (to species if possible), deer, empty, people, pigs, rats, mice, and cats. We estimated that we would need approximately 1,000 photos/category to train the convoluted neural network we are using for this project (D. Morris, pers. commun., Norruzedeh et al. 2018). Because our cameras captured few photos of deer, cats, and pigs, we are collaborating with other groups who have focused camera traps on these species to obtain more training data.

The photo-labeling process was time-consuming and labor-intensive in the Windows software we were using. In October 2019, NRDS created a desktop photo sorting application, which greatly eased the burden of viewing and labeling photos. The application was connected directly to the database where the camera deployments, checks, and removals were recorded. Users simply selected a folder containing camera trap photos they intended to sort and the corresponding camera check in the database. Animal observation records were then created in the same database whenever a person assigned a photo or set of photos to a specific taxon. The photo sorting application also stored the folder of photos as "not reviewed," "pending," and "complete" to track sorting progress, and only uploaded the labeled photos of interest to our cloud-based storage system, which was important because this effort led to gigabytes of data.

This project consisted of partners based throughout Hawaii, Alaska, and Colorado, so we faced the common challenge with camera trapping projects of sharing large amounts of data among different institutions (Newey et al 2015). Despite technology-forward intentions, we often resorted to saving the images to external hard drives and physically mailing them, leading to device failure and lost time. An early plan to account for this amount of data and its path through the sorting and modeling process would have helped us be more efficient. We have moved toward cloud-based services, including Dropbox, Azure, Amazon, and Google, which facilitate data storage and use by multiple project partners, and initial development and subsequent re-training of the models. We communicated among users to ensure that the same photos were not viewed twice. The NRDS app's built-in feature of marking a folder's status as reviewed or not prevented redundancies, but only if the viewers did not change the folder or filenames. Among partners, we also had to ensure that we could reduce the amount of observer bias in interpreting the image data (Meek et al. 2015). All partners viewing photos needed to be able to consistently identify the animals in the photos to the appropriate taxa grouping, which proved challenging at first (e.g., rats versus mice or birds to species).

**Algorithm Development**

Here we offer a short summary of lessons learned while developing a neural network model to sort our photo data. We followed the lead of Norouzzadeh et al. (2018), who applied these techniques to a large camera trap dataset from the African Serengeti, building on the work of Gomez-Villa et al. (2017). These models, and others that have sought to refine them (e.g., Tabak et al. 2018, Falzon et al. 2020, Weideng et al. 2020), have focused on classifying large mammals in grasslands or open forests, so we needed to adapt them for small animals in tropical rainforests.

Neural network models comprise a branch of machine learning in which computers can "learn" to do tasks without being explicitly programmed to do them. The subfamily of models we used (deep and convoluted neural networks) learn the project-specific objective by fitting complex and multilayered numerical algorithms to training data that show the computer the input and the desired output with a labeled data set. In this case, the training data was the sorted and tagged photos, and the output was the animal captured in them (e.g., rat, bird, human, pig, or empty). To fit the algorithm, we first converted the photo into its underlying numerical data (three layers on each of the red, green, and blue color spectra were assigned a numerical value for each pixel in each image). The neural network was trained by running a randomly selected subset of the photos through the algorithm, checking which predictions were right or wrong, and then adjusting the numerical connections between the layers accordingly.

Our initial goal was to develop a model to classify photos into animals of interest (i.e., birds and rodents) versus animals we were not interested in for this project (e.g., humans, pigs, cats, deer) and empty frames with only a trap. Hoping to avoid some of the laborious manual photo sorting required to generate training datasets, we approached Microsoft AI for Earth, which had developed an object detector model that sorts "animals of interest" from empty frames. In our photos, the model needed to find animals while ignoring the traps and the ever-changing illumination of plants, logs, and other objects in the forest understory. Although the existing Microsoft model can remove recurring objects (e.g., rodent traps) in photo frames, and is trained to classify species (https://github.com/microsoft/SpeciesClassification), it did not work well on small cryptic rodents and birds in the forest, because it was designed for large mammals in open ecosystems. We therefore manually sorted photos to generate our own set of project-specific training data, which we intend to share with Microsoft so they can improve their existing object detector and species classifiers.

It is relatively easy to train a model using the freely available tutorials provided by the Google/Tensorflow team (https://www.tensorflow.org/tutorials), but the field is rapidly growing (e.g., Tabak et al. 2018, Weideng et al. 2020). Even a year ago, it was challenging to read the photos into the numerical array format used by the models, but this step has since been streamlined by new packages like keras.preprocessing (https://keras.io/) and tf.data (https://www.tensorflow.org/guide/data ). With so many new products entering this field many it can be difficult to keep straight how the programs and packages interact with each other (https://www.tensorflow.org/guide/keras/overview).

We initially used R (https://www.r-project.org/) to manage photo data, convert photos into arrays, and develop the deep learning models using the keras package (https://keras.rstudio.com/). However, we soon switched to coding in Python, because that is the native programming language for Tensorflow. Currently, we are programming in R Studio (Version 1.2.5033), using R markdown [R version 3.6.0 (2019-04-26) "Planting of a Tree"], which allows us to hybridize our code, running R and Python in the same script to drive the Keras and Tensorflow programs.

Although these software programs are open source, managing them on a personal computing platform is sometimes problematic (but see Falzon et al. 2019 for some arguments in favor of using personal computers). On a desktop or virtual machine, all Keras and Tensorflow packages (also called dependencies in R and Python) must be loaded into the same computer, and the version that each of the other programs is expecting must be used. Cloud-based computing systems troubleshoot complex version issues with dependencies so that the underlying programs (Keras and Tensorflow) can actually develop the neural network models, saving considerable time for the modeler. A grant from Microsoft's AI for Earth program is allowing us now to use their cloud services on Azure in the coming months so we can focus on model development rather than software management.

Another consideration was getting our models into a format where they could be used by non-specialists (i.e., development of a user interface). There is a non-trivial role for software developers in this step, and we were fortunate to be working with partners at NRDS and Microsoft.

We have not yet completed the step of using neural network models to predict new data and sort our photos, due in part to its technical difficulty, and in part to the organizational and administrative hurdles we described above. We hope to have a complete set of training data by the end of July 2020 and to finalize the model development step and implement these models to sort photos in our data management pipeline in summer 2020.

**Initial Training Results and Validation**

The models we have trained to date performed well on validation data. We first trained a three-layer deep neural network model, which attained up to 97% accuracy in classifying images into rat or not rat. All failures of this model were false positives: it predicted that a rat was present when it was not. That said, these models are prone to overfitting the training data (i.e., memorizing all aspects of the photo, including the setting or branches of a repeatedly photographed tree) so these accuracy estimates we derived early in the process on limited data were likely an exaggeration of actual model accuracy. Overfitting to the training data can be checked by comparing the accuracy and loss curves for the training data and the validation data, which the modeler withholds from the training step but checks predictions against during the modeling process, as the model progresses through its learning epochs. If these curves are very far apart in their predictions (e.g., the model is predicting with 90% accuracy on the training data and only 70% on validation data on which it has not been trained), the model is likely overfit to the training data. The modeler can address overfitting by increasing the amount or complexity of the training data or by introducing regularization parameters into the model, like the dropout layers discussed below.

Next, we developed a convoluted neural network classifier to recognize animals in camera trap images (i.e., human, pig, deer, bird, rat, mouse, empty). Initial tests resulted in approximately 70% accuracy on validation data. We improved model fit by adding two dropout layers at a value of 30%. Dropout layers reduce the propensity of neural network models to overfit the training data, by removing a pre-defined percentage of the connections between the layers in the network (https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout?hl=zh-cn). We tried several options, with the model performing best at 30%. This step improved model accuracy from an initial approximately 70% to between 88% and 94%, depending on the run-input data varied with each model run due to our randomization procedures with the input photographs. We also tried several different model optimizers (i.e., Adam, stochastic gradient descent, and RMSprop) and had the best results with the Adam optimizer (Kingma and Lei Ba 2017). Katib (https://github.com/kubeflow/katib) enables users to test different model hyperparameters like changing optimizers and varying dropout layers and their rates for neural network models systematically and reproducibly. We have not used Katib or tools like this yet, so our success with the Adam optimizer and a 30% dropout rate were *ad hoc*. Other tools are constantly being developed by the machine learning community for the purpose of profiling neural network models and hyperparameter tuning.

## CONCLUSIONS

Camera traps are a powerful and seemingly easy-to-use tool that can efficiently record information not readily observable by humans. However, without careful planning for each phase of a camera trapping project (i.e., camera deployment; photo storage, sharing, and storing; and data analysis), a lot of time can be wasted capturing and managing unusable photos (e.g., of the wrong object), unidentifiable photos (e.g., because the storage card or storage folder was not properly labeled or tracked), or photos that take up storage space but are never viewed. Thus, it is important to think many of these issues through in the office prior to embarking on the project, plan for the costs of data management and storage, and model development while funding the project, and incorporate pilot studies in easily-accessible sites with lots of animal activity to perfect technicians' techniques of camera positioning given project objectives. Proper training for technicians can also reduce false positives and the amount of data to be managed.

In our experience in a wet, inaccessible environment, the money we spent on more expensive cameras was well spent, in terms of image quality, sensitivity of the motion sensor, and durability. The tradeoff was that we could afford fewer cameras. We compensated for the limited number of cameras by moving them often to capture the variation among traps, thus increasing our effective sample size. Moving cameras frequently cost field technician time but had an added advantage of ensuring that cameras were working and focused on the correct object.

It is critical to answer the following data management questions before beginning: Where will the photos be stored? How will we sort them? Will we store all the data, or only the useful data? How will we prevent mix-ups of SD cards? How will we track camera locations? How will we back up our data? How will we share it? Do we want to integrate AI into our data management pipeline and if so, how? What kind of statistical models do we want to use to analyze the sorted photos?

In our case, we had our Database Assistant (EMG) take a key role in this project, so our data flow was centralized. Although her position is primarily office-based, she understood the preliminary field work so understood all the potential obstacles. The involvement of partners with different expertise in coding and databases was a game-changer. Working with the NRDS database was another critical decision that greatly improved many stages of data management from camera location, to SD card deployed, to photo sorting and backup. NRDS may also prove to be a great user interface for the AI once it is fully developed. As discussed, cloud-based storage for photos and software also has many advantages but can be costly. If all these aspects are considered prior to camera deployment, camera trapping can be used to great advantage to enhance pest control and management.


## ACKNOWLEDGMENTS

We thank Justin Hite, Tyler Winter, Katherine Lauck, Bryn Webber, and Abigail Kreuser for help in the field and feedback on camera deployments. Stephen Miller with USFWS and Dan Morris with Microsoft AI for Earth provided expert advice and encouragement for algorithm development. Sam Aruch assisted with data management, software development, and photo review in NRDS software and provided insight on implementation of a user interface for the AI. Funding was provided by the U.S. Fish and Wildlife Service, State of Hawaii Department of Land and Natural Resources, and American Bird Conservancy. Automatic Trap Company loaned some cameras for this study. The Pacific Studies Cooperative Unit provided administrative and logistical support.


## LITERATURE CITED


Atkinson, C. T., R. B. Utzurrum, D. A. LaPointe, R. J. Camp, L. H. Crampton, J. T. Foster, and T. W. Giambelluca. 2014. Changing climate and the altitudinal range of avian malaria in the Hawaiian Islands: an ongoing conservation crisis on the island of Kaua'i. Global Change Biology 20:2426-2436.

Crampton, L. H., K. W. Brinck, K. E. Pias, B. A. P. Heindl, T. Savre, J. S. Diegmann, and E. H. Paxton. 2017. Linking occupancy surveys with habitat characteristics to estimate abundance and distribution in an endangered cryptic bird. Biodiversity and Conservation 26:1525-1539.

Falzon G, C. Lawson, K-W. Cheung, K. Vernes, G. A. Ballard, P. J. S. Fleming, A. S. Glen, H. Milne, A. T. Mather-Zardain, and P. D. Meek. 2020. ClassifyMe: a field-scouting software for the identification of wildlife in camera trap images. Animals 10(1):58.

Gomez-Villa, A., A. Salazar, and F. Vargas. 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.

Hammond, R. L., L. H. Crampton, and J. T. Foster. 2015. Breeding biology of two endangered forest birds on the island of Kaua'i. Condor 117:31-40.

Hammond, R. L., L. H. Crampton, and J. T. Foster. 2016. Nesting success of native and introduced forest birds on the island of Kauai. Journal of Avian Biology 47:252-262.

Hawai'i DLNR (Department of Land and Natural Resources). 2015. Hawai'i's state wildlife action plan. Prepared by H. T. Harvey and Associates, Honolulu, HI.

Kingma, D. P., and J. Lei Ba. 2017. Adam: a method for stochastic optimization. arXiv:1412.6980v9. 30 Jan 2017. https://arxiv.org/pdf/1412.6980.pdf

Leonard, D. L. 2008. Recovery expenditures for birds listed under the US Endangered Species Act: the disparity between mainland and Hawaiian taxa. Biological Conservation 141: 2054-2061.

Meek, P. D., G.-A. Ballard, and P. J. S. Fleming. 2015. The pitfalls of wildlife camera trapping as a survey tool in Australia. Australian Mammalogy 37(1):13-22.

Newey, S., P. Davidson, S. Nazir, G. Fairhurst, F. Verdicchio, R. J. Irvine, and R. van der Wal. 2015. Limitations of recreational camera traps for wildlife management and conservation research: a practitioner's perspective. Ambio 44(4): S624-S635.

Norouzzadeh, M. S., A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Klune. 2018. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences U.S.A. 115(25):E5716-E5725.

Paxton E. H., M. Laut, J. P. Vetter, and S. J. Kendall. 2018. Research and management priorities for Hawaiian forest birds. Condor 120(3):557-565.

Rovero, F., F. Zimmermann, D. Berzid, and P. Meek. 2013. "Which camera trap type and how many do I need?" A review of camera features and study designs for a range of wildlife research applications. Hystrix, the Italian Journal of Mammalogy 24(2):148-156.

Tabak, M. A., M. S. Norouzzadeh, D. W. Wolfson, S. J. Sweeney, K. C. VerCauteren, N. P. Snow, J. M. Halseth, P. A. Di Salvo, J. S. Lewis, M. D. White, B. Teton, J. C. Beasley, P. E. Schlichting, R. K. Boughton, B. Wight, E. S. Newkirk, J. S. Ivan, E. A. Odell, R. K. Brook, P. M. Lukacs, A. K. Moeller, E. G. Mandeville, J. Clune, and R. S. Miller. 2018. Machine learning to classify animal species in camera trap images: applications in ecology. Methods in Ecology and Evolution 10(4):585-590.

Trolliet, F., M.-C. Huynen, C. Vermeulen, and A. Hambuckers. 2014. Use of camera traps for wildlife studies. a review. Biotechnology, Agronomy, Society and Environment 18(3): 446-454.

Tweed, E. J., J. T. Foster, B. L. Woodworth, W. B. Monahan, J. L. Kellerman, and A. Lieberman. 2006. Breeding biology and success of a reintroduced population of the critically endangered Puaiohi (*Myadestes palmeri*). Auk 123:753-763.

USFWS. 2006. Puaiohi (*Myadestes palmeri*). 5-year review: summary and evaluation. U.S. Fish and Wildlife Service, Pacific Islands Fish and Wildlife Office, Honolulu, HI.

VanderWerf, E. A., L. H. Crampton, P. K. Roberts, J. S. Diegmann, and D. L. Leonard. 2014. Survival estimates of wild and captive-released Puaiohi, an endangered Hawaiian thrush. Condor 116:609-618.

Weideng W., G. Luo, J. Ran, and J. Li. 2020. Zilong: a tool to identify empty images in camera-trap data. Ecological Informatics 55:101021.