Methods in Ecology and Evolution

PATRICK C GRAY (Orcid ID : 0000-0002-8997-5255)

Article type : Research Article

Handling editor: Oscar Gaggiotti

A Convolutional Neural Network for Detecting Sea Turtles in Drone Imagery

Authors

Patrick C. Gray[1], Abram B. Fleishman[2], David J. Klein[2], Matthew W. McKown[2], Vanessa S. Bézy[3], Kenneth J. Lohmann[3], and David W. Johnston[1]

Affiliations

[1]Division of Marine Science and Conservation, Nicholas School of the Environment, Duke University Marine Laboratory, Beaufort, North Carolina, USA

[2]Conservation Metrics, Inc. Santa Cruz, California, USA

[3]Department of Biology, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, USA

Correspondence

Patrick Gray, Division of Marine Science and Conservation, Nicholas School of the Environment, Duke University Marine Laboratory, 135 Duke Marine Lab Rd, Beaufort, NC 28516, USA

Email: patrick.c.gray@duke.edu

Abstract (max 350 words)

1. Marine megafauna are difficult to observe and count because many species travel widely and spend large amounts of time submerged. As such, management programs seeking to conserve these species are often hampered by limited information about population levels.
2. Unoccupied aircraft systems (UAS, aka drones) provide a potentially useful technique for assessing marine animal populations, but a central challenge lies in analyzing the vast amounts of data generated in the images or video acquired during each flight. Neural networks are emerging as a powerful tool for automating object detection across data domains and can be applied to UAS imagery to generate new population-level insights. To explore the utility of these emerging technologies in a challenging field setting, we used neural networks to enumerate olive ridley turtles (*Lepidochelys olivacea*) in drone images acquired during a mass-nesting event on the coast of Ostional, Costa Rica.
3. Results revealed substantial promise for this approach; specifically, our model detected 8% more turtles than manual counts while effectively reducing the manual validation burden from 2,971,554 to 44,822 image windows. Our detection pipeline was trained on a relatively small set of turtle examples (N=944), implying that this method can be easily bootstrapped for other applications, and is practical with real-world UAS datasets.
4. Our findings highlight the feasibility of combining UAS and neural networks to estimate population levels of diverse marine animals and suggest that the automation inherent in these techniques will soon permit monitoring over spatial and temporal scales that would previously have been impractical.

Keywords: convolutional neural networks, deep learning for ecology, marine megafauna, marine population monitoring, object detection, sea turtles, unoccupied aircraft systems

1 | INTRODUCTION

Accurate and efficient population estimates are crucial for ecological studies and wildlife management (Cohen et al., 2003; Krebs, 1978). For many marine megafauna species, these data are difficult to collect because the animals spend much of their time under water, move rapidly over large areas, and occupy remote habitats. As a result, aerial surveys are commonly used to collect population data for these largely inaccessible species, and in recent years researchers have turned to unoccupied aircraft systems (UAS, or drones) for these tasks (Johnston, 2019). Surveying populations using UAS can be less logistically challenging than traditional methods, and can also reduce costs and human risk (Arona et al., 2018) without sacrificing data quality (Hodgson et al., 2018; Johnston et al., 2017). Such surveys have been successfully undertaken with a number of animals, including dugongs (Hodgson et al., 2013), seals (Johnston et al., 2017; Seymour et al., 2017), sea turtles (Sykora-Bodie et al., 2017), and several seabird species (Hodgson et al., 2016).

Globally, six of the seven marine turtle species are listed on the IUCN Red List of Threatened Species under various categories of extinction risk. Estimating the abundance of sea turtle populations is important for conservation efforts, as is developing robust estimates of density in specific breeding, foraging, and nesting areas where negative interactions may occur (James et al., 2005). This may be especially true for species like olive ridley sea turtles that exhibit mass-nesting, and which aggregate in extraordinarily dense concentrations in coastal areas. While UAS-based methods can facilitate these population assessments (Rees et al., 2018), an essential part of surveys is analyzing the resulting images and videos to determine the number of turtles present. Until now, analyses of this type have typically been carried out by trained observers who carefully view each image and count the number of animals present (Sykora-Bodie et al., 2017), but because these analyses are time-consuming and labor-intensive, they place a significant constraint on UAS surveys.

One possible way to overcome problems with analyzing images from drones is to automate methods for the detection, localization, and enumeration of target animals. Computer vision techniques have the potential to greatly increase the efficiency, repeatability, and precision of image assessments and overcome bottlenecks posed by large imagery datasets (Weinstein, 2017). Indeed, a variety of computer vision and machine learning techniques have been applied to assess wildlife populations using data collected not only by UAS imagery (Seymour et al., 2017), but also by camera traps (Schneider et al., 2018; Weinstein, 2018), traditional aerial imagery (Chabot et al., 2018), and satellites (Fretwell et al., 2014; Lynch and Schwaller, 2014; Moxley et al., 2017).

Several of these studies have applied modern object-based image analysis and conventional machine learning methods, but interest in deep-learning techniques for more complex detection of specific objects within images and video has been growing. Convolutional neural networks (CNNs), a prominent category of deep-learning classifier inspired by the neural connections in the human brain, are a fundamental source of recent computer vision advances and allow efficient discrimination of objects in noisy and complex environments (Lecun et al., 2015). Although CNN models have typically been applied to large-scale computer vision and image recognition problems, such as efficiently differentiating millions of images into thousands of classes of objects from standardized image libraries such as ImageNet (Krizhevsky et al., 2012), they have also been applied to other domains such as image and video data collected for ecological analysis.

Camera traps have been an early testbed for CNNs applied to ecological data. Camera traps are easy to operate and generate high-resolution imagery, but typically collect many unwanted frames due to false camera triggering. CNNs have shown considerable ability to detect objects of interest, such as birds and mammals, from these sources (Gupta and Verma, 2018; Schneider et al., 2018; Yousif et al., 2018). Progress has also been made in CNN-based detection of animals in UAS imagery. For example Borowicz et al (Borowicz et al., 2018) successfully counted Adélie penguins using a CNN (Szegedy et al., 2015). Beyond enumeration, CNN approaches can be used to identify between many species. One recent study was capable of differentiating nearly 600 common North American bird species with only 4% error rates in classification (Van Horn et al., 2015). Additionally,

these methods can potentially be applied in the acoustic realm, inasmuch as a CNN exists for monitoring a population of bats through automated detection of their echolocation signals (Aodha et al., 2018). The reduction in required human effort from applying these systems can be considerable. Norouzzadeha et al (2017) found their CNN could identify animals in 99.3% of the 3.2 million image Snapshot Serengeti dataset with the same accuracy as their crowdsourced identifications, saving volunteers the equivalent of 8.4 years of human labeling effort.

Although promising, CNNs can be prohibitively complex to implement. Moreover, they are computationally intensive and may require more data than is practical for most ecological studies. For example, Merlin used upwards of 50,000 images generated through human annotation to distinguish among bird species (Van Horn et al., 2015). Although details vary widely across studies, and although mitigation strategies such as transfer learning do exist, the application of these techniques to real-world problems clearly poses significant technical challenges.

In the present study, we explored the feasibility of using sophisticated yet accessible deep-learning techniques to increase the efficiency of an aerial-image-based population assessment for sea turtles. Specifically, we used a CNN to detect and enumerate olive ridley sea turtles in UAS-generated imagery from at-sea surveys conducted during a mass-nesting event in Ostional, Costa Rica. To our knowledge, this is the first use of CNNs for detecting sea turtles in aerial imagery and demonstrates the broad applicability of combining UAS-based data collection with neural networks for monitoring populations of marine animals.

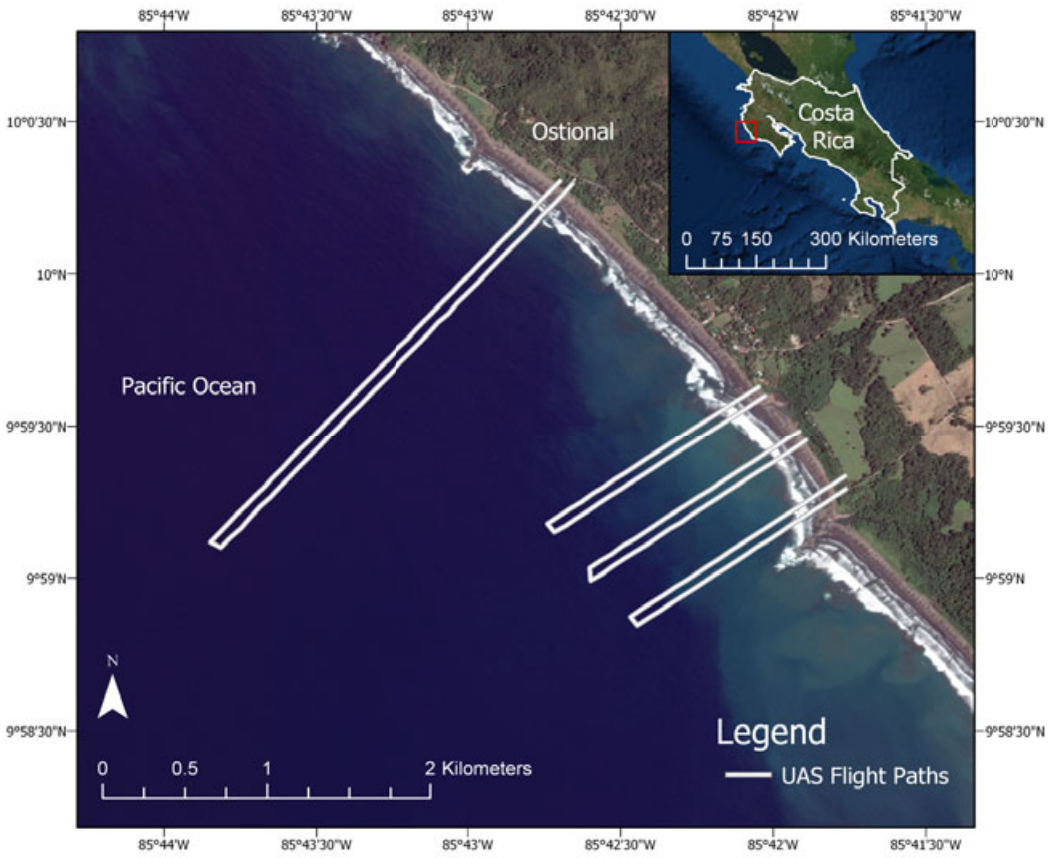


Figure 1. Map of the study site at Ostional, Costa Rica with an overview of unoccupied aircraft system (UAS) flight paths. Imagery Sources: Esri, DigitalGlobe

2 | MATERIALS AND METHODS

2.1 | Study Area

At-sea surveys of marine turtles were conducted in nearshore (< 3 km from land) waters of the Pacific within the marine protected area at the Ostional National Wildlife Refuge on the Nicoya Peninsula of Costa Rica (Figure 1). The refuge extends 200 meters inland from the high tide line and approximately 5.5 km offshore. Mass-nesting events of olive ridley sea turtles occur at Ostional Beach almost every month of the year. Peak nesting season coincides with the rainy season (May–November).

2.2 | UAS Imagery and Collection

Aerial surveys were conducted using an eBee (senseFly SA) fixed wing UAS, a modular UAS constructed of light-weight foam and powered by a single electric motor in push configuration (Sykora-Bodie et al., 2017). The UAS was outfitted with a Canon PowerShot S110 near-infrared (NIR) camera to capture aerial photographs. Initial tests with NIR and traditional RGB imagery revealed that NIR imagery provides superior contrast that facilitates detection of turtles in surface waters. Images were collected during transects designed for estimating turtle densities in nearshore waters (Figure 1, and see Sykora-Bodie et al., 2017). Flights were conducted opportunistically during daylight hours, regardless of tidal state or sun angle. A total of 20 UAS flights were conducted along four transects perpendicular to the beach (five flights per transect) during August 6, 7, 8, and 9, generating a series of overlapping false-color near-infrared (NIR) jpeg images (N=1059, 12.1 megapixel, pixel dimensions = 4048 x 3048).

2.3 | Image processing and human counts of turtles

To generate the dataset of labeled turtle locations, 467 of the 1,059 UAS images were used. Using iTag (https://sourceforge.net/projects/itagbiology/; version 0.6), three independent reviewers tagged turtles in the image set, taking approximately six hours for each reviewer to go through all of the images (Sykora-Bodie et al., 2017). A subset of these manually reviewed images (N=275) were used for model training and the remainder (N=192) were used for direct comparison of counts between manual review and CNN detection. As described previously in Sykora-Bodie et al (2017), reviewers used pre-set identification criteria to assign each possible turtle into one of two categories, “certain” or “probable”. Certain turtles were those in which appendages were visible and definitive identification was possible. Probable turtles were objects that resembled turtles in size, shape, and color, but could not be identified with certainty. For each photograph, the number of certain and probable turtles was determined. The location (x,y pixel coordinates) of each known or putative turtle within the image was also recorded in iTag. The GPS coordinates of each image were collected in

WGS84 and were transformed into a Universal Transform Mercator Zone 16N projection, which is a square grid with constant distances in meters. Pixel locations in each image were converted to UTM coordinates for each turtle detection. After counts were completed turtle detections were compiled into a text file with (x, y) coordinates of the center of each identified object. The subset used for training had N=616 certain turtle labels and N=328 probable turtle labels. The subset used for comparison between manual counts and CNN detection had N=384 certain turtle labels and N=253 probable turtle labels. For training and validation, data from these two categories were grouped together into a single class to facilitate automated assessments of detection.

2.4 | Deep Learning Model

An overview of the workflow applied to imagery acquired by the UAS is provided in Figure 2a. Briefly, the imagery data set was first partitioned into two components. One was manually labeled while the other was reserved for testing the model. The labeled data was used to train and validate the CNN and assess its initial performance. Once the CNN was trained, the testing data was run through the CNN to detect and enumerate turtles imaged during drone flights. Aspects of the CNN deployed in this workflow are visually expanded in figures 2b and 2c below.

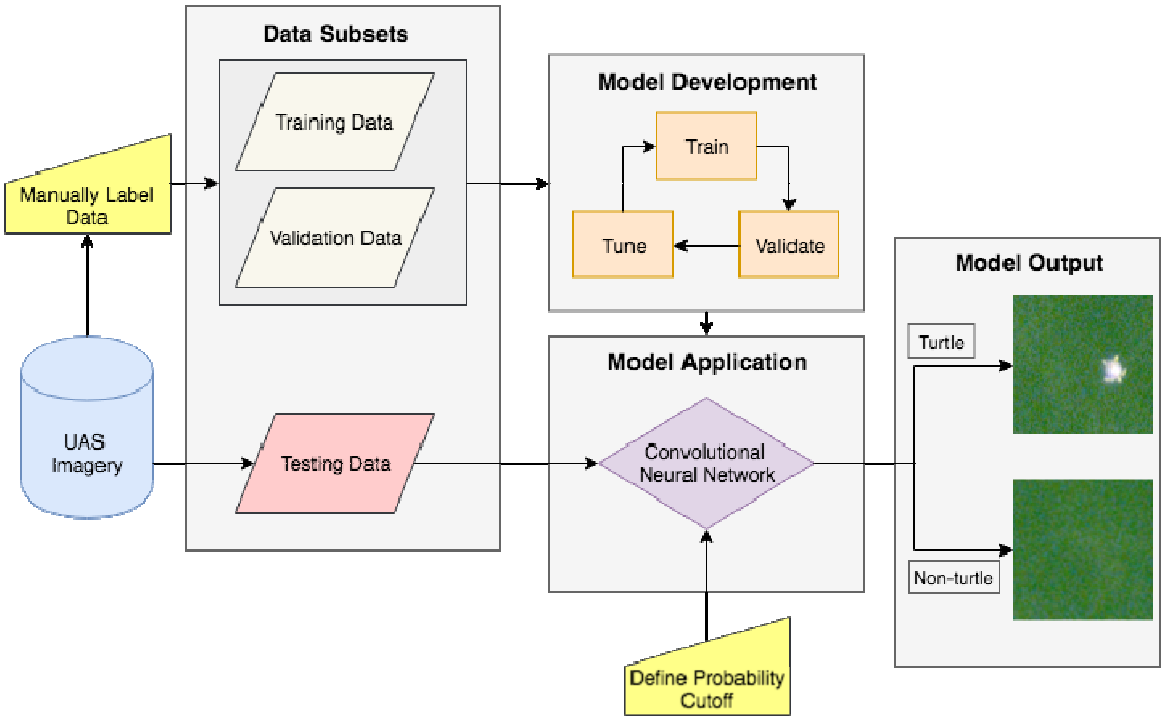


Figure 2a. Overview of Convolutional Neural Network development, application and output as applied to drone imagery of olive ridley turtles in the coastal waters of Ostional, Costa Rica.

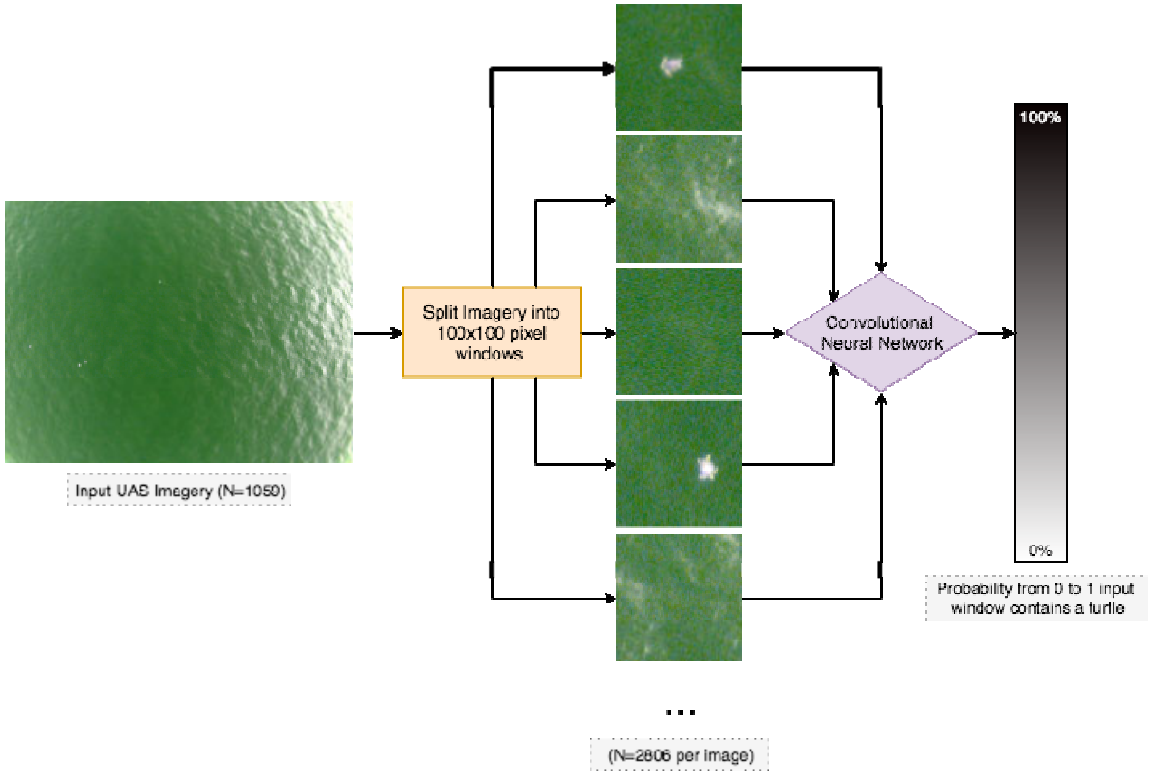


Figure 2b. Overview of convolutional neural network (CNN) "Model Application" from Figure 2a. UAS images were subdivided into 100 x 100 pixel windows, which served as input to the CNN. The CNN then computed a probability that each window included a turtle.

2.5 | Data Input and Cleaning

A visual inspection of turtles in the images revealed that each turtle could fit within a 50 x 50 pixel spatial region. Based on this observation each image was decomposed into an array of 100 x 100 pixel windows, with 1/3 overlap in the x and y directions (N=2806 windows per image). The overlap was chosen so that turtle centers would fall within or near the interior region of only one window, to minimize instances of missed or duplicated turtle detections because of window centering. A binary classification approach was used in which these 100 x 100 pixel windows were denoted with the positive class (1) if the central 50 x 50 region contained the center of a turtle. If a window did not contain a turtle in the center, it was denoted with the negative class (0).

2.6 | Neural Network Architecture and Training

Each training example for the CNN was comprised of a 100 x 100 x 3 tensor (x by y by Green, Red, Near Infrared) and the accompanying label classifying the example as negative or positive (0 or 1, respectively). Examples of the positive class (N=944) were assembled by cropping the images around each labeled (x+dx, y+dy) turtle coordinate, where the variables (dx, dy) have a uniform random distribution between -25 and +25. Thus, the positive examples all had turtle centers randomly distributed in the interior 50 x 50 pixel spatial region. This was done to simulate the random positioning of turtles within windows extracted from new images in deployment. Although there were no instances of multiple turtles within the center of a 50 x 50 pixel region in the training data, this approach would count multiple turtles in that 50 x 50 pixel region as one. Using a class balance of 10 negative examples for each positive example in our training data, examples of the negative class (N=9440) were assembled by cropping random 100 x 100 pixel windows from the image library, provided the center of the window was at least 125 pixels away from the center of a labeled turtle (in both the x and y directions). This led to a total of 10384 labeled examples. To train the CNN, a random 85% of the positive and negative examples were chosen as training data and the remaining 15% used as validation data.

Given the training data set was comprised of only N=8826 examples, a CNN of modest size was employed (Figure 2c). For detailed information on general CNN architecture and training, see Lecun et al. (2015) for a technical yet cogent overview. The CNN for this study was comprised of four convolutional layers with 64 3x3 kernels interleaved with max pooling layers. These convolutional layers form the backbone of the CNN and slide over the image essentially outputting heatmaps of various features within the image, called feature maps. Our CNN with 64 kernels will output 64 feature maps at each convolutional layer. The interwoven max pooling layers slide over the convolutional layer's feature maps with a 2x2 window and output the maximum value in that window. Important features from the maps are typically still retained by this simple max operation while the overall size of the feature maps is reduced with each iteration. Thus, the CNN keeps the same effective "field of view" while considerably reducing dimensionality. Including multiple iterations of

interleaved convolutional and max pooling layers, developing a smaller feature map each run-through, permits a CNN to ingest noisy and variable images, find useful features within them, and condense that into a relatively small yet informative final feature map. The first layer of a CNN typically creates maps of features such as edges, curves, and color gradients. The feature maps created in deeper layers in a CNN are more abstract and aggregate the previous layer's feature maps; in our case, combining them into groups of curves and edges that may indicate turtle flippers or shells. Through this process, the CNN extracts the distinguishing features that will permit effective classification. This process of feature extraction was followed by two fully connected layers of 1024 neurons. In order to prevent overfitting of the model, 50% of the connections between the neurons of these two fully connected layers were randomly ignored (dropout ratio of 0.5) during training.

The fully connected layers take the final 64 feature maps, ideally representing useful and high-level image components, and learn a mapping from those feature maps to the output classes (turtle – non-turtle). We used a binary normalized exponential (softmax) function output layer, which takes the final, fully connected layer and provides a continuous value between 0 and 1 for each window. Values closer to 1 indicate a higher likelihood that a turtle is present in the central 50x 50 region of the window. The CNN was trained to optimize the performance of the classification model (via a categorical cross-entropy, or log loss function) using stochastic gradient descent (SGD), with a learning rate of 0.01 and a Nesterov momentum of 0.9. Training was continued over 20 epochs, after which the model parameters were selected from the epoch resulting in the lowest loss on the validation set of N=1558 (N=153 of which contained turtles).

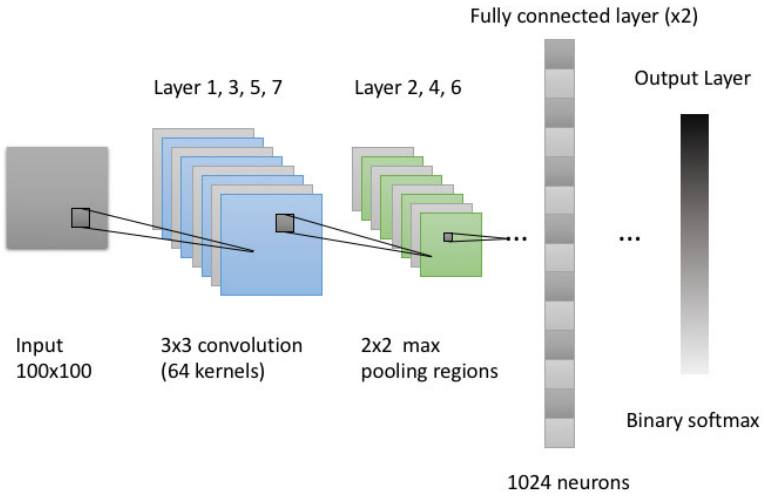


Figure 2c. Overview of the convolutional neural network (CNN) architecture. The CNN had four convolutional layers alternating with max pooling layers, these layers perform the feature extraction for the CNN, effectively distinguishing which aspects of the image are informative for classification. These layers were followed by two fully connected layers of 1024 neurons which combine the previously extracted features into meaningful combinations that ideally provide some predictive power for classification. The final layer employed a binary normalized exponential (softmax) function which ingests the final fully connected layer and its learned combinations of features, and returns a value between 0 and 1, with higher values signaling higher confidence of a turtle in the image window.

2.7 | Model Validation

Detections were reviewed with an internally developed custom user interface, allowing quick review of the 100x100 pixel windows, sorted by the CNN's output turtle likelihood (Figure 3). The tool uses a variety of keyboard shortcuts to note and correct instances of duplicate detections, false positives, or any false negatives encountered.

Because the windows overlapped by 30%, we detected some turtles more than once. When there was more than one detection of a turtle, we selected the window that included a larger proportion of the turtle or the turtle closest to the center of the photo. Double detections were removed manually by sorting true detections in photo capture order and marking duplicates with a new label.

Figure 3. Screenshot of the custom application used to rapidly review turtle detections.

3 | RESULTS

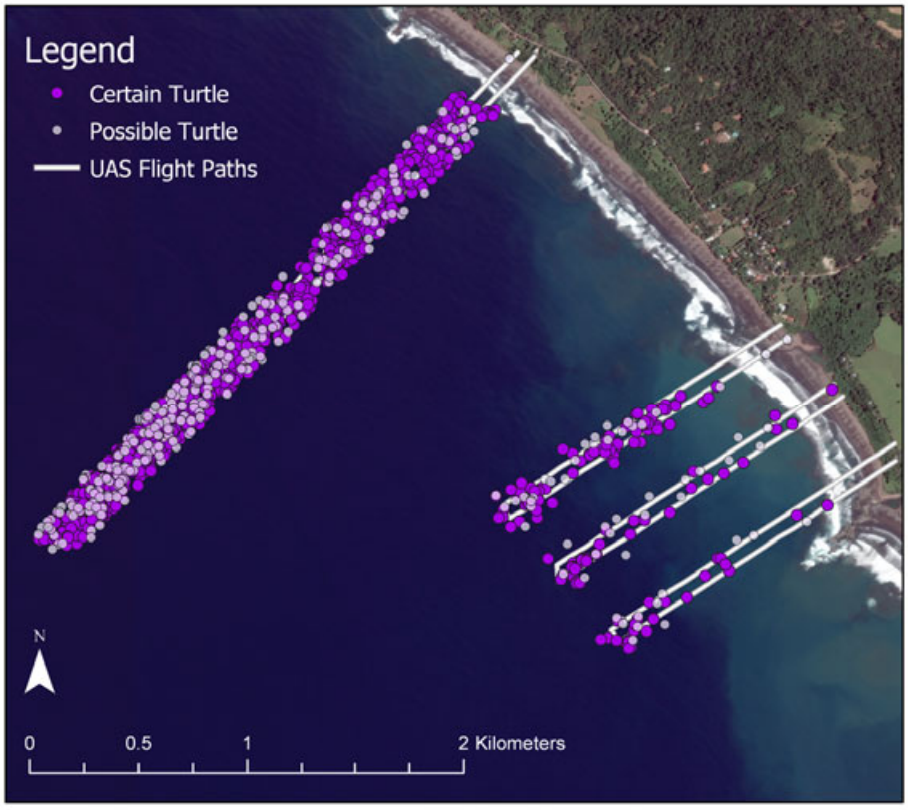


Figure 4. Distribution of certain and probable turtle detections from the convolutional neural network after the output was manually validated. Unoccupied aerial system (UAS) flight paths show distribution of turtle detections along survey transects.

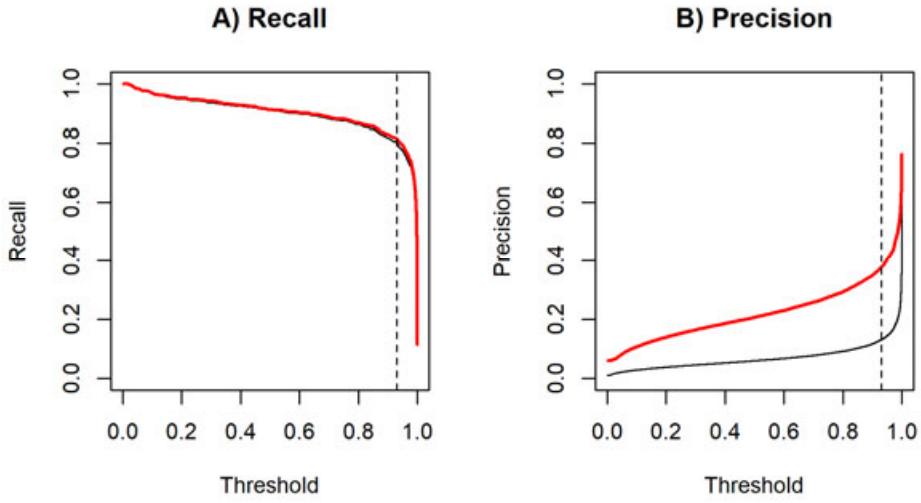
3.1 | Model Performance

The overall accuracy of our model to detect turtles was 99.83%, however, overall accuracy is not the best metric in classification problems with a sparse class of interest such as our study species, where the vast majority of image windows do not contain turtles. Instead we present recall and precision as metrics of model performance. Recall describes the number of true positives from the model divided by the total number of true positives in the data, or simply, what proportion of true positives were detected. Precision describes the number of true positives from the model divided by the total number

of detections by the model, again more simply, what proportion of detections were actually correct. After training and validation, the model was tuned to prioritize recall in order to miss as few true positives as possible, at the cost of additional false positives, with a detection probability threshold of 0.93. Out of 2,971,554 total windows (1,059 images each broken into 2,806 windows of 100x100 pixels), at this threshold the model flagged 6,696 windows as having turtles. Of those 6,696, we manually reviewed all windows and found there were 874 with certain turtles, 218 with probable turtles, and 5,402 with no turtles (Figure 4). Recall and precision metrics were calculated by combining the certain and probable turtle counts (Table 1). Accuracy, precision, and recall were quantified without manually verifying all 2,971,554 windows by reviewing all windows with a detection probability of 0.93-0.05 (N=44,799). This lower bound was determined empirically, while turtles could still exist in windows below that number, we found negligible true positives below a probability threshold of 0.15 and made the assumption that the total number of turtles detected, as well as our precision and recall statistics, wouldn't change materially with further review below the 0.05 probability threshold. This assumption prevented the necessity to manually review the remaining 2,926,732 windows below the 0.05 threshold. At this lower threshold 199 additional certain turtles and 137 probable turtles were found, and 44,463 windows with no turtles. Assuming these are all of the turtles within our dataset, this leads to 16.3% precision and 76.5% recall when using the 0.93 probability threshold for this model (Figure 5). The recall for our testing dataset closely matched the validation data (Figure 5A) while precision was poorer in the testing dataset (Figure 5B) suggesting slight model overfitting.

Table 1. Confusion matrix for turtle detection results when the model is tuned to show detections above the 0.93 confidence threshold. Shaded cells represent windows that were correctly classified (true positives and true negatives). *Validated non-turtle numbers are based on manually verifying windows down to a 0.05 probability threshold.

		Predicted	
		Turtle	**Non-turtle**
Validated	**Turtle**	1092	336
	Non-Turtle	5,604	2,964,522*



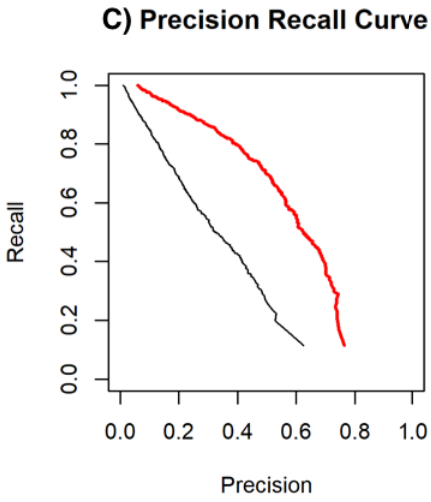


Figure 5. Convolutional Neural Network (A) recall, (B) precision, and (C) precision-recall curve for training (red) and full testing (black) data. Recall is the number of true positives from the model divided by the total number of true positives in the data. Precision is the number of true positives from the model divided by the total number of detections by the model. These two values represent model performance. Our model was tuned to optimize recall at the cost of precision.

3.2 | CNN vs Manual Counts

Using the same review method that generated the training data, a comparison of verified CNN detections to manual counts shows similar results but marginally better detection capability, approximately 8-9%, for both certain and probable turtle detections with the CNN (Table 2).

Table 2. A comparison of manual turtle counts, detections from the Convolutional Neural Network (CNN), and the percent difference from manual count to the CNN-based method.

	Manual	CNN	% Difference
Certain	384	418	+8.9
Probable	253	274	+8.3
Total	637	692	+8.6

3.3 | Model Validation Effort

It took one hour of analyst time to review detections from the 0.93 threshold (6,696 windows). It took an additional 11.8 hours to review all detections between probability thresholds of 0.93 and 0.05 (44,799 windows).

4 | DISCUSSION

Our study represents the first use of deep learning methods to assess at-sea densities of sea turtles. Results of the CNN analysis are similar to manual counts of the same imagery in a previous density assessment (Sykora-Bodie et al., 2017). The CNN approach identified 8-9% more turtles (Table 2), suggesting previous assessment should be viewed as conservative. Overall, our results demonstrate the feasibility of using neural networks to facilitate the analysis of images acquired for the purpose of monitoring animal populations. The general approach described here can also be applied to aerial surveys for other species large enough to be detected in coastal or pelagic marine environments. The present study illustrates how CNNs can facilitate efforts to enumerate sea turtles in drone imagery and eliminate analyst biases introduced during manual counts.

In many conservation studies, researchers must balance the risks of committing type I and type II errors. Most statistical analyses of scientific data focus on reducing type I errors in an effort to maximize the ability to reject null hypotheses with confidence. In conservation biology, however, the consequences of committing a type II error are often far worse (Shrader-Frechette 1994). Thus, researchers typically adopt approaches that minimize the likelihood of making type II errors. When applying a machine learning approach to counting species at risk, researchers can tune the system to minimize either Type I or Type II errors by balancing recall (the number of true positives from the model divided by the total number of true positives in the data) and precision (the number of true positives from the model divided by the total number of detections by the model).

In the context of this study, our model reduced the manual analysis burden to 1.5% of the initial amount using a 0.93 probability threshold. This was achieved without a diminished ability to detect turtles, in fact the model identified more turtles in a subset of the imagery than were identified using manual counts. While many deep-learning systems must reach predictably high precision and recall (e.g. detecting a stoplight in a vision system for a self-driving car), models applied to ecological data can be tuned to achieve alternative outcomes and can be useful even when they have low precision. Our model probability threshold was intentionally set low, minimizing the likelihood of committing type II errors (i.e., failing to detect a turtle when it is present) when assessing images. Here, we accept the financial cost of marginally more analyst time to review detections in an effort to generate the most robust density information for conservation purposes.

Although our model's precision is likely acceptable for many conservation applications, our testing data yielded lower precision than did the training and validation data (Figure 5). This difference is likely due to some overfitting of the CNN on the training data, perhaps caused by the limited number of training samples and differences between the training and testing datasets, which were captured on different days, under different weather and sun angle conditions, and contained some land images which were not in the training dataset. CNNs are prone to overfitting in circumstances of limited training data and if the network is too large for the given relationships it is attempting to model (Srivastava et al., 2014). Our training sample class imbalance (10:1 negative to

positive), while improving precision and recall on the validation set, may have also decreased precision in our testing data. A larger training dataset with examples of more varied image conditions might help mitigate this issue. The main source of false detections in this study were large jellyfish, breaking waves, and glare. Planning flights to reduce glare (e.g., during periods of lower sun angle), as well as image processing prior to CNN application, could substantially reduce these errors. For this study, our goal was to build a robust CNN, able to function well under variable conditions, in order to avoid imposing additional constraints on drone flights.

Major benefits of convolutional neural networks include: (1) they can be applied across images in different conditions, and (2) the model can be repeatedly upgraded using additional training data (Oquab et al., 2014). Thus, when new imagery is acquired during future deployments, the performance of the model can be improved by following the same initial process (lowering the probability threshold and manually inspecting detections), then adding the additional images of verified turtles into a new, larger training dataset. This iterative approach is likely to be useful across many similarly noisy and imbalanced datasets, increasing precision, improving detection of rarer image variants, and saving considerable time relative to the brute-force manual inspection of all data.

The next iteration of this study's CNN could be trained to identify false positive generating classes in addition to the primary class of interest. This has been shown to force the neural network to better separate these objects within its internal representation and thus reduce false positives (unpublished data, A. B. Fleishman). While not an issue in our study, adjustments should be made to permit detection of multiple turtles within close proximity, given that our current model would count multiple turtles within the 50 x 50 pixel region as a single turtle, in order to increase usefulness on data with denser aggregations or mating behavior. Beyond these two relatively minor changes, further understanding how fine-tuning a pre-trained CNN compares to our network built from scratch will be beneficial. Fine-tuning a CNN is the process of beginning with a pre-trained model, trained on a large dataset such as ImageNet (Jia Deng et al., 2009) or the Common Objects in Context dataset (Lin et al., 2014), and then adding in additional training samples specific to the desired detection problem. This process, more generally called transfer learning, has been shown to allow fewer training samples

(Razavian et al., 2014) and reduce overfitting (Yosinski et al., 2014), thus increasing the potential applicability of this method in real world biology scenarios (Schneider et al., 2018). If the CNN from our study were first trained on ImageNet, which would build out many of the class agnostic image process capabilities, such as edge detectors, and then trained on our turtle dataset, we hypothesize there would be a noticeable improvement in precision. A final avenue of inquiry is comparison of this custom designed CNN architecture to models currently available open-source and "off the shelf" that provide state of the art results in other domains (Lin et al., 2017; Ren et al., 2017).

Open source CNN implementations, combined with transfer learning, will allow CNNs to be improved upon with additional data and fed back into the community, considerably enhancing our ability to detect and study wildlife. Improvements in CNN speed and efficiency may eventually permit detectors to run in real-time on UAS (Huang et al., 2017), facilitating autonomous monitoring and behavioral analysis. In order to build out these capabilities, we advocate for the creation of appropriately sized open-access training datasets of aerial imagery for all sea turtle species in various conditions, permitting rapid creation and improvement of CNNs for sea turtle population monitoring globally.

Acknowledgments

We thank Seth Sykora-Bodie for providing details on his study on UAS-based sea turtle population assessment. Thanks to Everette Newton for conducting flights and contributing to manual counts. Thanks to Walter Wright and Matt Elardo for manual counting of turtles. We greatly appreciate SINAC and employees at the Ostional National Wildlife Refuge for logistical support and our research being permitted by the Costa Rican government. This research was funded by the National Science Foundation grant (IOS-1456923) to K. J. Lohmann.

Author Contributions: MWM and DWJ conceived the ideas and designed methodology; VSB and KJL collected the data; ABB and DJK analyzed the data; PCG led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Data Accessibility: The authors have uploaded the convolutional neural network code to Github.com at url: https://github.com/patrickcgray/cnn_sea_turtle_detection and DOI: 10.5281/zenodo.1973808. All relevant imagery, including training and testing labels, has been stored on the Dryad data repository: DOI: 10.5061/dryad.5h06vv2.

References

Aodha, O.M., Gibb, R., Barlow, K.E., Browning, E., Firman, M., Freeman, R., Harder, B., Kinsey, L., Mead, G.R., Newson, S.E., Pandourski, I., Parsons, S., Russ, J., Szodoray-Paradi, A., Szodoray-Paradi, F., Tilova, E., Girolami, M., Brostow, G., Jones, K.E., 2018. Bat detective—Deep learning tools for bat acoustic signal detection. PLoS Comput. Biol. 14. https://doi.org/10.1371/journal.pcbi.1005995

Arona, L., Dale, J., Heaslip, S.G., Hammill, M.O., Johnston, D.W., 2018. Assessing the disturbance potential of small unoccupied aircraft systems (UAS) on gray seals (*Halichoerus grypus*) at breeding colonies in Nova Scotia, Canada. PeerJ 6, e4467. https://doi.org/10.7717/peerj.4467

Borowicz, A., McDowall, P., Youngflesh, C., Sayre-Mccord, T., Clucas, G., Herman, R., Forrest, S., Rider, M., Schwaller, M., Hart, T., Jenouvrier, S., Polito, M.J., Singh, H., Lynch, H.J., 2018. Multi-modal survey of Adélie penguin mega-colonies reveals the Danger Islands as a seabird hotspot. Sci. Rep. 8, 1–9. https://doi.org/10.1038/s41598-018-22313-w

Chabot, D., Dillon, C., Francis, C.M., 2018. An approach for using off-the-shelf object-based image analysis software to detect and count birds in large volumes of aerial imagery 13.

Cohen, J.E., Jonsson, T., Carpenter, S.R., 2003. Ecological community description using the food web, species abundance, and body size. Proc. Natl. Acad. Sci. U. S. A. 100, 1781–6. https://doi.org/10.1073/pnas.232715699

Fretwell, P.T., Staniland, I.J., Forcada, J., 2014. Whales from space: Counting southern right whales by satellite. PLoS One 9, 1–9. https://doi.org/10.1371/journal.pone.0088655

Gupta, P., Verma, G.K., 2018. Wild Animal Detection Using Deep Convolutional Neural Network, in: Proceedings of 2nd International Conference on Computer Vision & Image Processing. Springer Singapore, pp. 327–338. https://doi.org/10.1007/978-981-10-7898-9_27

Hodgson, A., Kelly, N., Peel, D., 2013. Unmanned aerial vehicles (UAVs) for surveying Marine Fauna: A dugong case study. PLoS One 8. https://doi.org/10.1371/journal.pone.0079556

Hodgson, J.C., Baylis, S.M., Mott, R., Herrod, A., Clarke, R.H., 2016. Precision wildlife monitoring using unmanned aerial vehicles. Sci. Rep. 6. https://doi.org/10.1038/srep22574

Hodgson, J.C., Mott, R., Baylis, S.M., Pham, T.T., Wotherspoon, S., Kilpatrick, A.D., Raja Segaran, R., Reid, I., Terauds, A., Koh, L.P., 2018. Drones count wildlife more accurately and precisely than humans. Methods Ecol. Evol. 9, 1160–1167. https://doi.org/10.1111/2041-210X.12974

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017–Janua, 3296–3305. https://doi.org/10.1109/CVPR.2017.351

James, M.C., Ottensmeyer, C.A., Myers, R.A., 2005. Identification of high-use habitat and threats to leatherback sea turtles in northern waters: New directions for conservation. Ecol. Lett. 8, 195–201. https://doi.org/10.1111/j.1461-0248.2004.00710.x

Jia Deng, Wei Dong, Socher, R., Li-Jia Li, Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conf. Comput. Vis. Pattern Recognit. 248–255. https://doi.org/10.1109/CVPRW.2009.5206848

Johnston, D.W., 2019. Unoccupied Aircraft Systems in Marine Science and Conservation. Ann. Rev. Mar. Sci. 11, annurev-marine-010318-095323. https://doi.org/10.1146/annurev-marine-010318-095323

Johnston, D.W., Dale, J., Murray, K.T., Josephson, E., Newton, E., Wood, S., 2017. Comparing occupied and unoccupied aircraft surveys of wildlife populations: assessing the gray seal (Halichoerus gryus) breeding colony on Muskeget Island, USA. J. Unmanned Veh. Syst. 5, 178–191. https://doi.org/10.1139/juvs-2017-0012

Krebs, C., 1978. Ecology: The Experimental Analysis of Distribution and Abundance, New York: Harper and Row. https://doi.org/10.2307/3545

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Adv. Neural Inf. Process. Syst. 1–9. https://doi.org/http://dx.doi.org/10.1016/j.protcy.2014.09.007

Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P., 2017. Focal Loss for Dense Object Detection, in: Proceedings of the IEEE International Conference on Computer Vision. pp. 2999–3007. https://doi.org/10.1109/ICCV.2017.324

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 8693 LNCS, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Lynch, H.J., Schwaller, M.R., 2014. Mapping the abundance and distribution of Adélie penguins using landsat-7: First steps towards an integrated multi-sensor pipeline for tracking populations at the continental scale. PLoS One 9. https://doi.org/10.1371/journal.pone.0113301

Moxley, J.H., Bogomolni, A., Hammill, M.O., Moore, K.M.T., Polito, M.J., Sette, L., Sharp, W.B., Waring, G.T., Gilbert, J.R., Halpin, P.N., Johnston, D.W., 2017. Google Haul Out: Earth Observation Imagery and Digital Aerial Surveys in Coastal Wildlife Management and Abundance Estimation. Bioscience 67, 760–768. https://doi.org/10.1093/biosci/bix059

Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M., Packer, C., Clune, J., 2017. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proc. Natl. Acad. Sci. 115. https://doi.org/10.1073/pnas.1719367115

Oquab, M., Bottou, L., Laptev, I., Sivic, J., 2014. Learning and transferring mid-level image representations using convolutional neural networks. Proc. IEEE. https://doi.org/10.1109/CVPR.2014.222

Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S., 2014. CNN features off-the-shelf: An astounding baseline for recognition. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. 512–519. https://doi.org/10.1109/CVPRW.2014.131

Rees, A., Avens, L., Ballorain, K., Bevan, E., Broderick, A., Carthy, R., Christianen, M., Duclos, G., Heithaus, M., Johnston, D., Mangel, J., Paladino, F., Pendoley, K., Reina, R., Robinson, N., Ryan, R., Sykora-Bodie, S., Tilley, D., Varela, M., Whitman, E., Whittock, P., Wibbels, T., Godley, B., 2018. The potential of unmanned aerial systems for sea turtle research and conservation: a review and future directions. Endanger. Species Res. 35, 81–100. https://doi.org/10.3354/esr00877

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Schneider, S., Taylor, G.W., Kremer, S.C., 2018. Deep Learning Object Detection Methods for Ecological Camera Trap Data, in: Computer Vision and Pattern Recognition.

Seymour, A.C., Dale, J., Hammill, M., Halpin, P.N., Johnston, D.W., 2017. Automated detection and enumeration of marine wildlife using unmanned aircraft systems (UAS) and thermal imagery. Sci. Rep. 7, 45127. https://doi.org/10.1038/srep45127

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J. Mach. Learn. Res. 15, 1929–1958. https://doi.org/10.1214/12-AOS1000

Sykora-Bodie, S.T., Bezy, V., Johnston, D.W., Newton, E., Lohmann, K.J., 2017. Quantifying Nearshore Sea Turtle Densities: Applications of Unmanned Aerial Systems for Population Assessments. Sci. Rep. 7. https://doi.org/10.1038/s41598-017-17719-x

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., Hill, C., Arbor, A., 2015. Going Deeper with Convolutions, in: Conference on Computer Vision and Pattern Recognition. pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S., 2015. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 07–12–June, 595–604. https://doi.org/10.1109/CVPR.2015.7298658

Weinstein, B.G., 2018. Scene-specific convolutional neural networks for video-based biodiversity detection. Methods Ecol. Evol. 9, 1435–1441. https://doi.org/10.1111/2041-210X.13011

Weinstein, B.G., 2017. A computer vision for animal ecology. J. Anim. Ecol. https://doi.org/10.1111/1365-2656.12780

Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural

networks?, in: Advances in Neural Information Processing Systems. pp. 1–9. https://doi.org/10.1109/IJCNN.2016.7727519

Yousif, H., He, Z., Kays, R., 2018. OBJECT SEGMENTATION IN THE DEEP NEURAL NETWORK FEATURE DOMAIN FROM HIGHLY CLUTTERED NATURAL SCENES, in: Proceedings - International Conference on Image Processing, ICIP. pp. 3095–3099. https://doi.org/10.1109/ICIP.2017.8296852