Received: 25 January 2018 | Accepted: 24 March 2018

DOI: 10.1111/2041-210X.13011

APPLICATION

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

Scene-specific convolutional neural networks for video-based biodiversity detection

Ben G. Weinstein

Department of Fisheries and Wildlife, Marine Mammal Institute, Oregon State University, Newport, OR

Correspondence
Ben G. Weinstein
Email: weinsteb@oregonstate.edu

Handling Editor: William Pearse

Abstract

1. Finding, counting and identifying animals is a central challenge in ecology. Most studies are limited by the time and cost of fieldwork by human observers. To increase the spatial and temporal breadth of sampling, ecologists are adopting passive image-based monitoring approaches. While passive monitoring can expand data collection, a remaining obstacle is finding the small proportion of images containing ecological objects among the majority of frames containing only background scenes.
2. I proposed a scene-specific convolutional neural network for detecting animals of interest within long duration time-lapse videos. Convolutional neural networks are a type of deep learning algorithm that have recently made significant advances in image classification.
3. The approach was tested on videos of floral visitation by hummingbirds. Despite low frame rates, poor image quality, and complex video conditions, the model correctly classified over 90% of frames containing hummingbirds. Combining motion detection and image classification can substantially reduce the time investment in scoring images from passive monitoring studies.
4. These results underscore the promise of deep learning to lead ecology into greater automation using passive image analysis. To help facilitate future applications, I created a desktop executable that can be used to apply pre-trained models to videos, as well as reproducible scripts for training new models on local and cloud environments.

KEYWORDS

automated monitoring, computer vision, hummingbirds, neural networks, remote cameras

1 | INTRODUCTION

Observing biodiversity is critical to understanding species ecology, demography and behaviour. Many animals are secretive and difficult to detect in nature. Direct observation is expensive and potentially dangerous to researchers. Without sufficient observations, it is difficult to make conservation decisions regarding animal presence, population size or habitat preferences. As an alternative to direct observation, many researchers are adopting passive monitoring systems using image capture (Anderson et al., 2016; Gregory, Carrasco Rueda, Deichmann, Kolowski, & Alonso, 2014; Schmid, Reis-Filho, Harvey, & Giarrizzo, 2016). While image-based monitoring reduces the effort and invasiveness of documenting animal presence, it generates a large number of extraneous images without animals (Swanson et al., 2015). Reviewing these “empty” images is laborious, tedious and non-reproducible.

Computer vision is a form of image-based artificial intelligence that mimics human vision by generating rules for the form, grouping and changes of image pixels (Weinstein, 2017). To find objects in video, computer vision tools use “Background Subtraction” to distinguish foreground and background pixels. By defining the image features for a background model, background subtraction can be used to filter out empty frames. The increase in accessibility of computer vision libraries (Bradski, 2000) has led to a hope for a single background subtraction method for diverse ecological systems (Bowley, Andes, Ellis-Felege, & Desell, 2016; Elias, Golubovic, Krintz, & Wolski, 2017; Price Tack et al., 2016; Weinstein, 2015). However, the complex backgrounds of marine and terrestrial environments, combined with the myriad forms of animal shape and patterning, make a universal detector unlikely. Instead, the computer vision community has made progress using convolutional neural networks with scene-specific training data (Babaee, Dinh, & Rigoll, 2017; Braham & Van Droogenbroeck, 2016). Convolutional neural networks are a machine learning approach that uses training data to determine the image features that best delineate image classes (LeCun, Bengio, & Hinton, 2015). Convolutional neural networks have a series of hierarchical layers that connect image pixels with higher order combinations of image features. Colloquially known as “deep learning”, neural networks are attractive to ecologists because image features are not apriori designed, but rather learnt from existing labelled images (Marburg & Bigham, 2016; Wilf et al., 2016). The goal of this paper was to test a scene-specific convolutional network to create an animal detector for time-lapse video.

Ecological studies often occur in a single location, with similar image backgrounds in recorded data. It may therefore be possible to build a scene-specific background model that can be applied to new videos or images in the same system (Bowley et al., 2016; Ren, Han, & He, 2013). My goal is to explore a pipeline for video-based biodiversity observation, describe its strengths and limitations, and outline avenues for further exploration. Previous work in ecological computer vision has used motion detection to screening images based on temporal filtering (Swinnen, Reijniers, Breno, & Leirs, 2014), pixel models (Weinstein, 2015), and edge detection (Price Tack et al., 2016). I build on background subtraction tools introduced in Weinstein (2015) to add deep learning classification of potential movement objects. I apply this model to a large dataset of videos typical of ecological surveys. These videos include dramatic changes in light, poor image quality, inclement weather, human disturbance and low frame rates. The aims of this paper are to (1) determine whether data trained from a subset of videos can be used as a background model for new videos taken from the same ecosystem, (2) compute the recall and precision of the model in finding target animals and (3) categorise remaining misclassified frames to direct future work.

2 | MATERIALS AND METHODS

2.1 | Approach

In 2015, I designed MotionMeerkat, a windows executable for computing background subtraction (Weinstein, 2015). MotionMeerkat uses preset thresholds for pixel change to screen images for novel objects. While this software has been applied to a diverse group of terrestrial and marine applications, it has three limitations, (1) the user must decide on the subtraction threshold, despite little intuition for ideal values, (2) the detector treats all motion objects equally, and makes no attempt at object classification, (3) the model performance is static, there is no mechanism to improve performance with new data. To address these challenges, I built DeepMeerkat, a tool that classifies potential motion objects using a convolutional neural network built with Google’s Tensorflow library (Abadi et al., 2016) (Figure 1). DeepMeerkat operates without preset values, classifies objects based on training data, and can be improved over time with newly labelled data.

DeepMeerkat is an extension of MotionMeerkat (Weinstein, 2015), with the addition of a neural network to classify movement objects as either foreground or background. Rather than train a novel network from scratch, I fine-tuned Google’s Inception V3 model

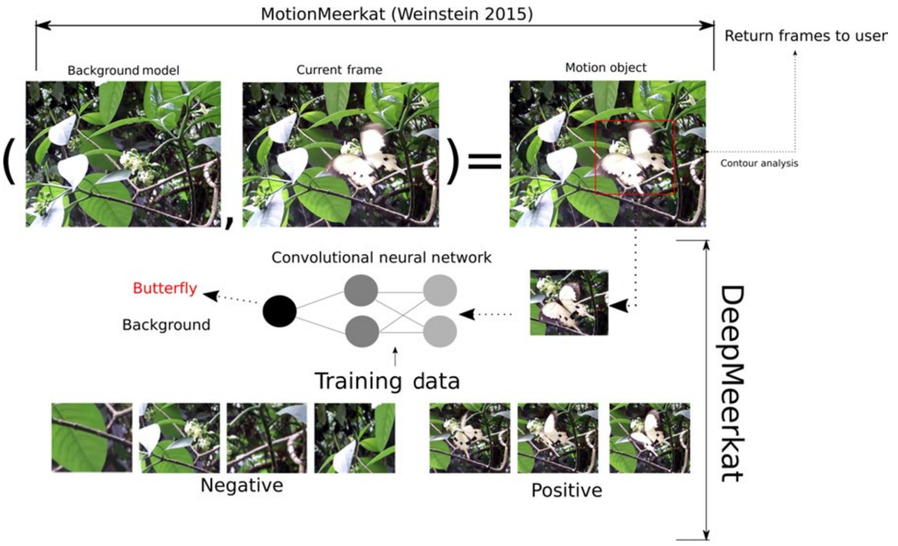


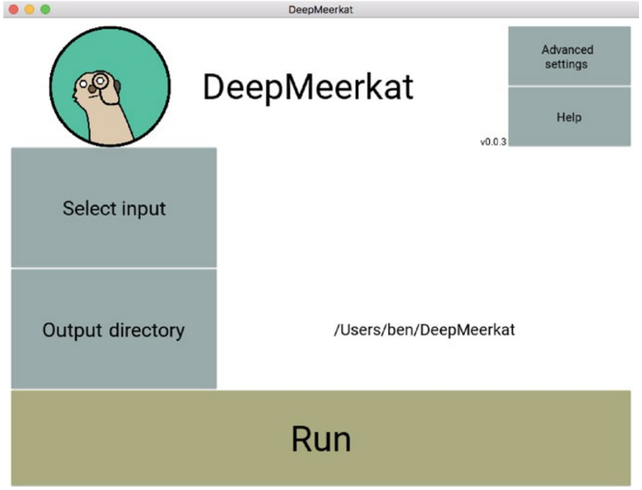
FIGURE 1 The proposed system to detect animals in long duration videos. This work extends Weinstein (2015) to include convolutional neural networks trained with scene-specific training data. In this figure, the network is trained on known images of *Papilio dardanus*, as well as crops of the video background. By changing the training images in the positive and negative categories, the model can learn classes specific to a given ecological project

(Szegedy et al., 2015), which was originally trained on the 2012 ImageNet data (Deng et al., 2009). Fine-tuning uses a pre-trained network to freeze the lower layers of the architecture, but retraining the top layers to better fit the new dataset. This capitalises on previous training and improves performance when using a smaller set of training images (Nogueira, Penatti, & dos Santos, 2017). On top of the frozen inception layers, I added a 50% dropout layer, then retrained a fully connected layer and a softmax layer to predict object classes. Dropout is a form of regularisation that randomly removes a proportion of nodes to reduce overfitting (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). The fully connected layer reduces the vector of image features to the desired dimensionality of length two (foreground and background). The softmax layer normalises this vector into probabilities that sum to one across all classes. DeepMeerkat is designed to be conservative, with a high threshold for retaining frames (acceptance value = 0.1). This means that the model must be more than 90% confident that a frame does not contain a foreground object to assign a background label. This prioritises minimising false negatives at the potential expense of increasing false positives.

For training the fine-tuned neural network, I collected images for each class and trained with a batch size of 100 for 20,000 steps. To reduce training time, the feature vectors for the frozen layers were extracted in parallel using Google Cloud DataFlow. These features were then the basis for retraining the new fine-tuned layers. To fit the specifications of the pre-trained frozen layers, the bounding boxes from motion detection were resized into three channel arrays with height and width of 299 pixels. Following Zhang, He, Cao, and Cao (2016), aspect ratios of bounding boxes were not maintained when passing boxes to the neural network. Model performance was measured using true positive rate, true negative rate, and precision. A DeepMeerkat GUI (Figure 2, Figures S1 and S2) is available for download for Mac and Windows with the pre-trained hummingbird model. In addition, I provide reproducible scripts for local and Google cloud environments to allow users to train new models, which can then be used in the local software.

2.2 | Test dataset

My collaborators and I have been studying hummingbird ecology using time-lapse cameras in the Ecuadorian Andes since 2013 (Weinstein & Graham, 2017). Cameras turn on at dawn, off at dusk, and record a photo every second for up to 5 days. Cameras filming individual flowers capture hummingbirds in less than 1% of images. Hummingbird visits are rapid and rare, lasting 3 to 5 s, with only a handful of visits a day. To train the network, I collected 14,432 image crops containing hummingbirds and 14,432 crops containing background vegetation and sky. To validate the accuracy of the model, I selected 70 half-day videos that represented a range of challenging backgrounds. These videos were not used in the model training. Within these videos there 532 frames (1.4%) containing hummingbirds and 37,677 frames (98.6%) containing background. Previous analysis showed motion detection to be effective in finding the majority of hummingbird visitation events (Weinstein, 2015). For the purposes of this article, I assumed that all events are captured by motion detection and were passed to the neural network for classification. While quantitative benchmarks are needed to validate model performance, they provide a coarse description of the errors associated with incorrect classification. I therefore reviewed each misclassified frame and categorised its potential challenges. Given the varied quality of biodiversity images captured in the field, there could be several pitfalls including: poor image quality, bounding box errors and non-hummingbird foreground objects. Each challenge requires a different remedy, and will shape the direction of future work.



Output directory /Users/ben/DeepMeerkat

FIGURE 2 The front screen of the DeepMeerkat GUI. A user can select a file or directory of videos to screen using a pre-trained model. The path to the model is set under "Advanced settings"

3 | RESULTS

Feature extraction of the fixed inception layers completed in 1 hr and 26 min on 15 CPUs. Training of the new layers completed in 27 min on a single CPU. Model evaluation on the 70 test videos completed in 4 hr and 38 min on 30 CPUs with an average frame rate of 17 frames/s. On average, a video contained 545.84 candidate motion frames (2.5% of total frames) that were sent for classification by the neural network. For each of frame, the neural network returned a probability of background or foreground. Using a standard acceptance threshold of 0.5 to classify background, the model had a true positive rate of 89.3% and a true negative rate of 91.9% (Table 1). Using a more conservative 0.1 acceptance threshold, the model had a 95.7% true positive rate and a 76.1% true negative rate. The trade-off between recall and precision varied by acceptance value, with a faster decrease in background recall at low acceptance values (Figure 3). The majority of motion detection objects were well centred and had high foreground classification probabilities (Figure 4). Using the more conservative 0.1 acceptance value, the remaining misclassified frames (n = 24) had poor bounding box segmentation

TABLE 1 Recall and precision statistics for the scene-specific background model based on acceptance rate. True foreground rate is the proportion of frames with foreground objects returned to the user. True background rate is the proportion of non-returned frames that contained background objects. Precision is the proportion of returned images that contained desired foreground objects. Statistics are reported for two acceptance values. A traditional 0.5 value means that if the probability of background was greater than 50%, the frame was not returned to the user. A more conservative 0.1 acceptance value means that only images with a probability of background greater than 90% were not returned to the user

Acceptance value	True foreground rate (%)	True background rate (%)	Precision (%)
0.5	89.3	91.9	13.5
0.1	95.7	76.1	5.4

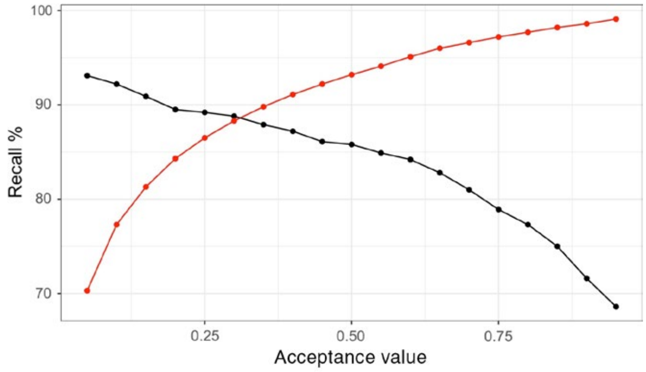


FIGURE 3 Recall curves (proportion of correct labels returned) for the hummingbird label (black) and background label (red) based on the threshold for acceptance value. For example, an acceptance value of 0.1 means that only frames that had a background probability of greater than 0.9 are labelled background

(*n* = 12), insufficient difference between background and training data (*n* = 1), strong changes in lighting (*n* = 7), and poor image quality (*n* = 4) (Figure 5). While these categories are subjective and not mutually exclusive, they provide important direction for avenues of future work.

4 | DISCUSSION

Deep learning holds the potential for revolutionary development in image-based biodiversity monitoring. Combining computer vision and neural networks, the proposed approach correctly returned 95.7% of frames containing hummingbirds, while ignoring 76.1% of frames containing only background objects. The combination of motion detection and neural network classification led to an enormous reduction in the number of frames needing human review, with often less than 2% of original frames requiring human annotation. These results confirm that convolutional neural networks can be retrained to specific ecological scenes and greatly reduce the time for video annotation.

A high proportion of the remaining misclassified frames were due to poor or incorrect bounding box segmentation. Of the 24 misclassified frames containing hummingbird images, 12 had bounding boxes that were not properly centred. When bounding boxes are too big, the hummingbird feature space is small compared to the background feature space, leading to poor classification performance. When bounding boxes are too small, they cut-off portions of the hummingbird, making the observed image less similar to the training data. This points to flaws in the motion detection rather than the classification network. The motion detection relies on a traditional mixture-of-Gaussian model (MOG) to generate an expected range of pixel values based on previous frames (Bouwmans, Gonzàlez, Shan, Piccardi, & Davis, 2014). Rather than separately computing motion detection and classification, the two steps could be combined by training a neural network on known bounding boxes, and using the entire frame for prediction (Ren, He, Girshick, & Sun, 2015). While this would eliminate the bounding boxes errors due motion detection, it would greatly reduce processing rates, since the network would need to evaluate every image.

Using a 0.1 acceptance threshold, the proportion of frames returned that contained foreground objects was 13.3%. This low precision is due to the conservative acceptance threshold, as was as the remaining challenges in classifying variable background objects. To solve this, the neural network needs more information to distinguish between the image classes. One option is to pass both the image crop and the corresponding crop in the background model to a network to score image similarity (Braham & Van Droogenbroeck, 2016). This approach computes a distance metric among images to delineate whether the current image crop is foreground or background (Zhang et al., 2016). While this kind of network would be more difficult to train, it could be more generalisable, since it is the differences between foreground and background that determine if the frames are returned to the user.

Future work must investigate the diversity of backgrounds and the size of image classes needed for model training. Similar tools are developing for screening camera trap images (Chen, Han, He, Kays, & Forrester, 2014), and finding organisms in remotely sensed imagery (Guirado, Tabik, Alcaraz-Segura, Cabello, & Herrera, 2017). Convolutional neural networks underlie many advances in these areas, and connecting insights across disciplines will be a key in developing best practices. While the promise of this technology has been well advertised (Pimm et al., 2015), the next stage is to establish the details. How many images per class? What is the diversity of objects that can be distinguished? While a single global background model for all ecological videos is unlikely to emerge, by training data on specific scenes we can reduce the burden of image review and increase automation in biodiversity data collection.

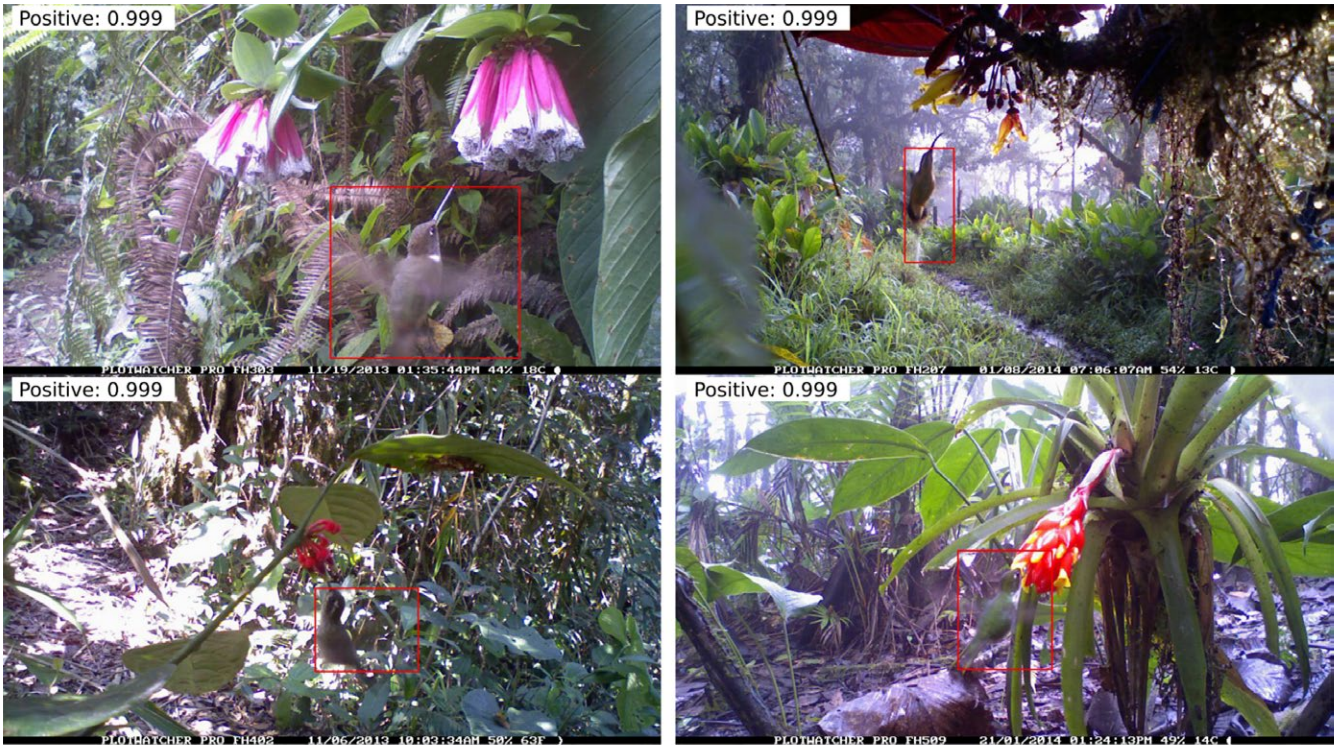

Positive: 0.999
PLOTWATCHER PRO FH303 11/19/2013 01:35:44PM 44% 18C
Positive: 0.999
PLOTWATCHER PRO FH207 01/08/2014 07:06:07AM 54% 13C
Positive: 0.999
PLOTWATCHER PRO FH402 11/06/2013 10:03:34AM 50% 63F
Positive: 0.999
PLOTWATCHER PRO FH509 21/01/2014 01:24:13PM 49% 14C

FIGURE 4 Positive detections of hummingbirds in long duration time-lapse videos. The bounding boxes were generated by motion detection and fed into a neural network to distinguish foreground ("positive") and background ("negative") objects. The probability of the predicted label, ranging from 0 to 1, is shown in red

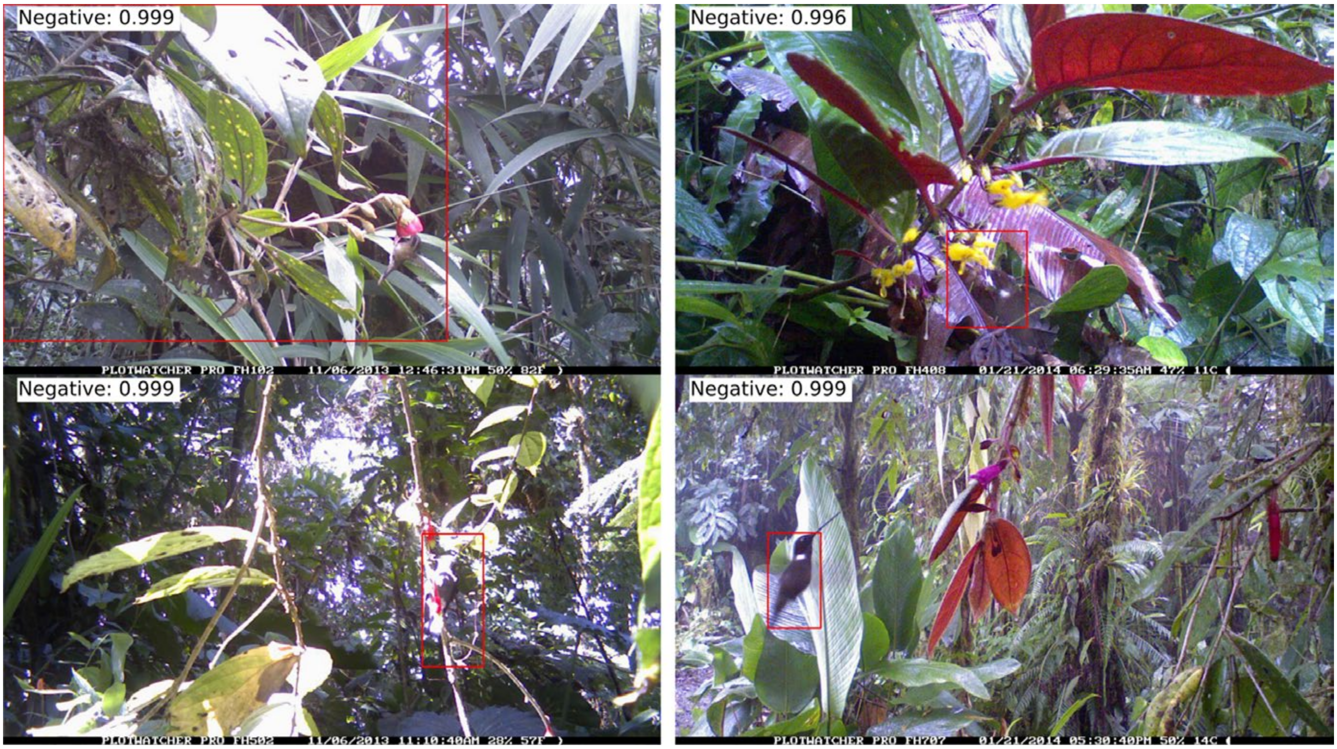

Negative: 0.999
PLOTWATCHER PRO FH102 11/06/2013 12:46:31PM 50% 82F
Negative: 0.996
PLOTWATCHER PRO FH408 01/21/2014 06:29:35AM 47% 11C
Negative: 0.999
PLOTWATCHER PRO FH502 11/06/2013 11:10:40AM 28% 57F
Negative: 0.999
PLOTWATCHER PRO FH707 01/21/2014 05:30:40PM 50% 14C

FIGURE 5 False negatives of hummingbirds in long duration time-lapse videos. The bounding boxes were generated by motion detection and fed into a fine-tuned neural network to distinguish foreground ("positive") and background ("negative") objects. The four selected images are symptomatic of the broader challenges, top left) inaccurate bounding box segmentation, top right) poor image quality, bottom left) rapid changes in lighting, bottom right) Incorrect classification of foreground objects

ACKNOWLEDGEMENTS

Thanks to Segment for supporting this work as part of their Open Data Fellowship. Thanks to the many Google engineers who shared code samples and answering questions on deploying machine learning models. The author declares no conflict of interest.

DATA ACCESSIBILITY

A DeepMeerkat executable for Mac and Windows is available with a default hummingbird model (http://benweinstein.weebly.com/deepmeerkat.html) (https://doi.org/10.5281/zenodo.1213328). All code for creating new models is available on github (https://github.com/bw4sz/DeepMeerkat). While it is recommended to build models using Google's cloud infrastructure for greatest speed and reproducibility, I have also provided guidelines on executing the supplied scripts using local resources (Appendix S1).

ORCID

Ben G. Weinstein http://orcid.org/0000-0002-2176-7935

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … Kudlur, M. (2016). TensorFlow: A system for large-scale machine learning. *OSDI*, *16*, 265–283.

Anderson, T. M., White, S., Davis, B., Erhardt, R., Palmer, M., Swanson, A., … Packer, C. (2016). The spatial distribution of African savannah herbivores: Species associations and habitat occupancy in a landscape context. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *371*, 20150314. https://doi.org/10.1098/rstb.2015.0314

Babaee, M., Dinh, D. T., & Rigoll, G. (2017). A deep convolutional neural network for background subtraction. *Pattern Recognition*, *76*, 635–649.

Bouwmans, T., Gonzàlez, J., Shan, C., Piccardi, M., & Davis, L. (2014). Special issue on background modeling for foreground detection in real-world dynamic scenes. *Machine Vision and Applications*, *25*, 1101–1103. https://doi.org/10.1007/s00138-013-0578-x

Bowley, C., Andes, A., Ellis-Felege, S., & Desell, T. (2016). Detecting wildlife in uncontrolled outdoor video using convolutional neural networks. In 2016 IEEE 12th International Conference on e-Science (e-Science) (pp. 251–259). IEEE. https://doi.org/10.1109/escience.2016.7870906

Bradski, G. (2000). The OpenCV library. *Doctor Dobbs Journal*, *25*, 120–126.

Braham, M., & Van Droogenbroeck, M. (2016). Deep background subtraction with scene-specific convolutional neural networks. International Conference on Systems, Signals, and Image Processing, 2016–June. https://doi.org/10.1109/iwssip.2016.7502717

Chen, G., Han, T. X., He, Z., Kays, R., & Forrester, T. (2014). Deep convolutional neural network based species recognition for wild animal monitoring. 2014 IEEE International Conference on Image Processing (ICIP), (pp. 858–862). https://doi.org/10.1109/icip.2014.7025172

Deng, J., Dong, W., Socher, R., Li, L-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, (June), (pp. 248–255). https://doi.org/10.1109/cvprw.2009.5206848

Elias, A. R., Golubovic, N., Krintz, C., & Wolski, R. (2017). Where's the bear? – Automating wildlife image processing using IoT and edge cloud systems. In *2017 IEEE/ACM second international conference on internet-of-things design and implementation* (IoTDI), Pittsburgh, PA, 2017 (pp. 247–258).

Gregory, T., Carrasco Rueda, F., Deichmann, J., Kolowski, J., & Alonso, A. (2014). Arboreal camera trapping: Taking a proven method to new heights. *Methods in Ecology and Evolution*, *5*, 443–451. https://doi.org/10.1111/2041-210X.12177

Guirado, E., Tabik, S., Alcaraz-Segura, D., Cabello, J., & Herrera, F. (2017). Deep-learning Versus OBIA for Scattered Shrub Detection with Google Earth Imagery: Ziziphus lotus as Case Study. *Remote Sensing*, *9*, 1220. https://doi.org/10.3390/rs9121220

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. https://doi.org/10.1038/nature14539

Marburg, A., & Bigham, K. (2016). Deep learning for benthic fauna identification. OCEANS 2016 MTS/IEEE Monterey, 1–5.

Nogueira, K., Penatti, O. A. B., & dos Santos, J. A. (2017). Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, *61*, 539–556. https://doi.org/10.1016/j.patcog.2016.07.001

Pimm, S. L., Alibhai, S., Bergl, R., Dehgan, A., Giri, C., Jewell, Z., … Loarie, S. (2015). Emerging technologies to conserve biodiversity. *Trends in Ecology & Evolution*, *30*, 685–696. https://doi.org/10.1016/j.tree.2015.08.008

Price Tack, J. L., West, B. S., McGowan, C. P., Ditchkoff, S. S., Reeves, S. J., Keever, A. C., & Grand, J. B. (2016). AnimalFinder: A semi-automated system for animal detection in time-lapse camera trap images. *Ecological Informatics*, *36*, 145–151. https://doi.org/10.1016/j.ecoinf.2016.11.003

Ren, X., Han, T. X., & He, Z. (2013). Ensemble video object cut in highly dynamic scenes. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (pp. 1947–1954). https://doi.org/10.1109/cvpr.2013.254

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Nips*, *39*, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Schmid, K., Reis-Filho, J. A., Harvey, E., & Giarrizzo, T. (2016). Baited remote underwater video as a promising nondestructive tool to assess fish assemblages in clearwater Amazonian rivers: Testing the effect of bait and habitat type. *Hydrobiologia*, *784*, 93–109. https://doi.org/10.1007/s10750-016-2860-1

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958. https://doi.org/10.1214/12-AOS1000

Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., & Packer, C. (2015). Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Scientific Data*, *2*, 150026. https://doi.org/10.1038/sdata.2015.26

Swinnen, K. R. R., Reijniers, J., Breno, M., & Leirs, H. (2014). A novel method to reduce time investment when processing videos from camera trap studies. *PLoS ONE*, *9*, e98881. https://doi.org/10.1371/journal.pone.0098881

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Frank Wang, Y. C. (2015). Going deeper with convolutions. In *2015 IEEE conference on computer vision and pattern recognition* (CVPR), Boston, MA, 2015 (pp. 1–9).

Weinstein, B. G. (2015). MotionMeerkat: Integrating motion video detection and ecological monitoring. *Methods in Ecology and Evolution*, *6*, 357–362. https://doi.org/10.1111/2041-210X.12320

Weinstein, B. G. (2017). A computer vision for animal ecology. *Journal of Animal Ecology*, *87*, 533–545. https://doi.org/10.1111/1365-2656.12780

Weinstein, B. G., & Graham, C. H. (2017). Persistent bill and corolla matching despite shifting temporal resources in tropical hummingbird-plant interactions. *Ecology Letters*, *20*, 326–335. https://doi.org/10.1111/ele.12730

Wilf, P., Zhang, S., Chikkerur, S., Little, S. A., Wing, S. L., & Serre, T. (2016). Computer vision cracks the leaf code. *Proceedings of the National Academy of Sciences of the United States of America*, *113*, 201524473.

Zhang, Z., He, Z., Cao, G., & Cao, W. (2016). Animal detection from highly cluttered natural scenes using spatiotemporal object region proposals and patch verification. *IEEE Transactions on Multimedia*, *18*, 2079–2092. https://doi.org/10.1109/TMM.2016.2594138

SUPPORTING INFORMATION

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

How to cite this article: Weinstein BG. Scene-specific convolutional neural networks for video-based biodiversity detection. *Methods Ecol Evol*. 2018;9:1435–1441. https://doi.org/10.1111/2041-210X.13011