Ecological Informatics 50 (2019) 191–196

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

An automated program to find animals and crop photographs for individual recognition

Patrick Buehler[a], Bill Carroll[a], Ashish Bhatia[a], Vivek Gupta[a], Derek E. Lee[b,*]

[a] Microsoft Azure, 1 Microsoft Way, Redmond, WA 98052, USA
[b] Department of Biology, Pennsylvania State University, University Park, PA 16802, USA

ARTICLE INFO

Keywords:
Animal detection
Capture-recapture
Computer vision
Histogram of oriented gradients
Image processing
Individual identification
Photo identification

ABSTRACT

Detailed data on individual animals are critical to ecological and evolutionary studies, but attaching identifying marks can alter individual fates and behavior leading to biases in parameter estimates and ethical issues. Individual-recognition software has been developed to assist in identifying many species from non-invasive photographic data. These programs utilize algorithms to find unique individual characteristics and compare images to a catalogue of known individuals. Currently, all applications for individual identification require manual processing to crop images so only the area of interest remains, or the area of interest must be manually delineated in each image. Thus, one of the main bottlenecks in processing data from photographic capture-recapture surveys is in cropping to an area of interest so that matching algorithms can identify the individual. Here, we describe the development and testing of an automated cropping program. The methods and techniques we describe are broadly applicable to any system where raw photos must be cropped down to a specific area of interest before pattern recognition software can be used for individual identification. We developed and tested the program for use with identification photos of wild giraffes.

1. Introduction

Computer vision applications have become important tools for ecological research (Weinstein, 2017). The proliferation of digital still and video camera traps (Burton et al., 2015; Rowcliffe and Carbone, 2008), and the use of digital photography as a primary source of individual-based data (Bolger et al., 2012; Moya et al., 2015) have greatly increased sampling, but our ability to process images remains a bottleneck in turning these data into ecological information (Weinstein, 2017). Several options exist for automated processing of camera trap data because the fixed-position mounting of camera traps allows computer vision applications to take advantage of the relatively unchanging background that occurs in every frame (Bradski, 2000; Price Tack et al., 2016; Swinnen et al., 2014; Weinstein, 2015). However, we are aware of no existing applications for processing data from photographic capture-recapture surveys where identification pictures are taken by an observer and the background of every image is different (but see Sherley et al., 2010).

Detailed data on individual animals from capture-recapture surveys are used in ecological and evolutionary studies to estimate demographic parameters such as rates of survival, reproduction, and movement (Lebreton et al., 1992; Williams et al., 2001). A common method for individual recognition of the animals is to apply a mark to the animal body in the form of a tag or other device with a unique code. However, attaching tags and other marks can alter individual fates and behavior leading to biases in the parameter estimates (McCarthy and Parris, 2004; Petersen et al., 2005; Wilson and McMahon, 2006), and creating ethical issues (Cuthill, 1991; May, 2004; Minteer and Collins, 2005). Consequently, there is increasing interest in using non-invasive methods for individual recognition such as photography of unique natural marks. Simultaneously, digital photography has led to substantial increases in sampling and a growing demand for automated procedures of photo-identification. Most photo-identification procedures require three steps. The first step is manual selection and/or cropping of an area of interest on the animal within the image; the second is an automated algorithmic comparison between the sample and a library of images which scores candidates by matching probability; and the final step is visual comparison of sample–candidate pairs to confirm positive matches.

Individual-recognition software has been developed to assist in identifying a diverse suite of species such as cheetahs (Kelly, 2001), elephants (Ardovini et al., 2008), tigers (Raj et al., 2015), salamanders (Gamble et al., 2008), fishes (Arzoumanian et al., 2005; Van Tienhoven et al., 2007), penguins (Sherley et al., 2010), and marine mammals

* Corresponding author.
E-mail address: derek@wildnatureinstitute.org (D.E. Lee).

https://doi.org/10.1016/j.ecoinf.2019.02.003
Received 22 February 2018; Received in revised form 1 February 2019; Accepted 2 February 2019
Available online 04 February 2019

(Adams et al., 2006; Gope et al., 2005). Flexible individual identification tools applicable to a wide range of species are also available (Bolger et al., 2012; Moya et al., 2015). These programs utilize algorithms to find unique individual characteristics and compare images to a catalogue of known individuals. Currently, all applications for individual identification require manual processing to crop images so only the area of interest remains (Arzoumanian et al., 2005; Bolger et al., 2012; Raj et al., 2015), or the area of interest must be manually delineated in each image (Kelly, 2001; Moya et al., 2015; Van Tienhoven et al., 2007). Thus, the main bottleneck in processing data from photographic capture-recapture surveys is in object detection for cropping or delineating an area of interest so that matching algorithms can identify the individual. Here, we describe the development and testing of an automated cropping program. The methods and techniques we describe are broadly applicable to any system where raw photos such as those obtained from field workers or citizen scientists must be cropped down to a specific area of interest before pattern recognition software is used for individual recognition. We developed and tested the program for use with identification photos of wild giraffes (*Giraffa camelopardalis*).

2. Material and methods

2.1. Object detection approach

We base our work on the well-known Histogram of Oriented Gradients (HOG) feature for object detection (Dalal and Triggs, 2005). HOG features capture both boundary edges and internal texture, and the contrast normalization they employ accounts for variation in lighting (see Fig. 1 for an example). Detecting objects in images has been attracting a lot of attention in the Computer Vision community. Commonly employed approaches are based on (i) Convolutional Neural Networks which automatically learn how to represent an object, and on (ii) approaches which use a hand-designed object representation. While deep-learned approaches have received increased attention ever since the popular AlexNet paper was published (Krizhevsky et al., 2012), it is worth pointing out that traditional methods such as HOG or SIFT (scale invariant feature transformation; Lowe, 1999) have been improved over many years and shown to work well on tasks such as people detection (Dalal et al., 2006; Dalal and Triggs, 2005), or as we show in this work, giraffe torso detection. These approaches are efficient to train and evaluate, do not require dedicated hardware such as an expensive graphics processing unit (GPU), and provide more insights into what the model learned.

There are similarities between HOG and the widely used SIFT descriptor. In both cases, orientation histograms are computed for an image grid to represent local image patches. The main difference is that HOG describes the whole image in a dense grid and at some particular scales, whereas SIFT computes multiple local image descriptors centered on automatically detected interest points. These interest points define not only a position in the image, but also a scale and an orientation which is typically used to make SIFT invariant to these transformations. Both HOG and SIFT perform normalization on the grid of histograms for an image patch (in HOG, this is based on blocks; in SIFT, it is performed for the whole grid) to improve invariance to changes in illumination.

The main steps to build our object detector were:

1) Collect a large set of images (see Section 3.1) and manually annotate each object-of-interest using a bounding box. These images are used for training and testing of the detection system.
2) Create crops of all annotated objects given by the bounding boxes from step 1; these are used as positive examples. In addition, create crops from image regions which do not show the object; these are used as negative examples (see Fig. 2).
3) Compute the HOG descriptor for each extracted crop. These serve as representations of the positive and negative crops.
4) Train a Support Vector Machine (SVM) classifier using the positive and negative HOG descriptors, as well as hard-negatives mined using an Active Learning approach. The trained SVM takes a single HOG descriptor of an arbitrary image region as input, and outputs a detection score which indicates if the region contains the object-of-interest.
5) Use the trained SVM to find (possibly none or multiple) occurrences of the object in new images. This is implemented by sliding a rectangular window over the image (typically left to right, top to bottom) and by evaluating the trained SVM at each window position to find all objects (see Section 2.3).

Details for model training (steps 2–4) are given in Section 2.2, and for model scoring (step 5) are given in Section 2.3.

2.2. Model training

As is the case for all supervised machine learning approaches, we require a set of training images to be provided where, in each image, all objects-of-interest are annotated. For object detection, these annotations are typically in the form of rectangles which are manually drawn around the objects. Given such annotations, our model can then be trained by following the steps 2–5 in the previous section. We will now provide more detail for each of these steps.

Step 2 – crop generation: We create crops of all objects given by the manually annotated bounding boxes from step 1; these are used as positive examples during model training and model evaluation. Negative examples are collected by (i) creating random crops from the same images, which do not overlap with any of the annotated objects,

Fig. 1. Visualization of a Histogram of Gradients (HOG) object detector (right) for a given image (left). HOG captures the dominant gradients of the image (e.g. the bicycle) while ignoring near uniform areas (e.g. the area in the foreground).

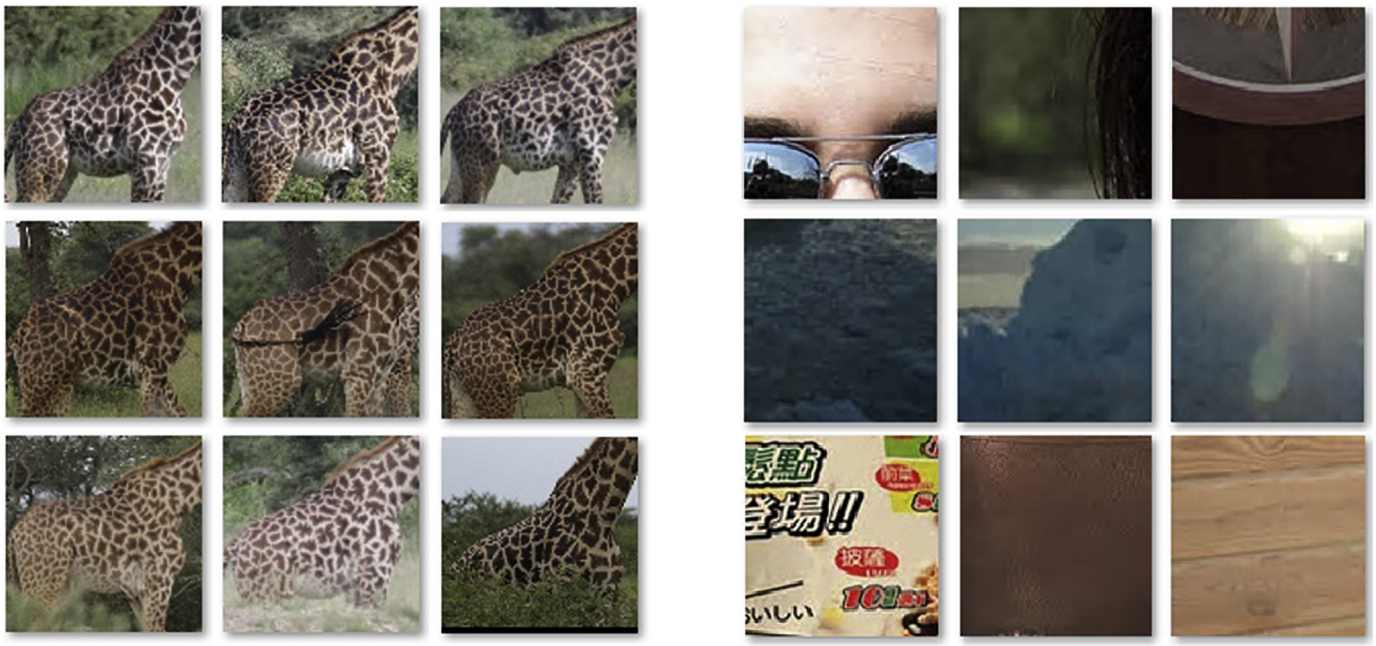

Fig. 2. Positive (left) and negative (right) examples used to train our giraffe detector.

and by (ii) creating random crops from out-of-domain images which are known not to contain the object-of-interest. See Fig. 2 for positive and negative crops.

Step 3 – HOG computation: In this step, the HOG descriptor is computed for each crop individually. In our implementation, this is a 4356-floating point vector. These vectors serve as representations of the crops which are shown to be more robust against imaging effects such as illumination or variations in background compared to the raw pixel values. The crops after computing the respective HOG descriptors are not needed anymore and can be discarded.

Step 4 – Support Vector Machine (SVM) training: We use a binary SVM to classify whether a crop contains the object-of-interest. The SVM takes the HOG descriptor of a crop as input, and outputs a detection score. To train the SVM, the positive and negative descriptors from step 3 are used.

Once trained, we noticed that the model could successfully localize the object in previously unseen images, however it often also misfired on unrelated regions. We therefore employed an Active Learning approach to iteratively find such misdetections, add them to the training set, and retrain the SVM. This approach to mine so-called hard negatives can be done fully automatically by using a large dataset of images which do not contain the object-of-interest, and hence all detections are guaranteed to be mistakes. With hard negatives added to the training set, the accuracy of the final model improved significantly (see Section 3.3).

2.3. Model scoring

Given the trained model from step 4, we can now build a system which finds (possibly multiple or none) objects in a given image. We use a sliding window approach for this task, where a rectangular region with fixed width and height is moved over the image, starting from the top left corner, to the bottom right (see Fig. 3). At each window location, the trained SVM is evaluated to obtain a score of the window containing the object-of-interest. All locations with scores above a certain threshold (by default this threshold is 0) are then used as object detections. In Fig. 3, only the green detection window (left) highlighted by the white arrow (right) is above this threshold.

This sliding is done independently at multiple scales since the relative size of the object in the image is typically unknown. Furthermore, instead of computing the HOG descriptor at each sliding window location, we use an efficient modification where the HOG descriptor is only computed once per image. The actual sliding is perform in the HOG space, by moving the window to the right or down with a stride length of one cell (see Section 2.4 for an explanation of a “cell”, and Section 2.5 for a popular computer vision library which implements this efficient search).

2.4. Histogram of oriented gradients descriptor

This section describes the HOG descriptor in more detail and explains how the descriptor is computed. Given an image, HOG computes local gradient orientation histograms, and then contrast-normalizes these local histograms over larger spatial regions, capturing not only boundary edges but also internal edges. The basic idea is that the appearance of an object can be characterized by the distribution of local intensity gradients and edge directions.

Fig. 4 shows the multiple stages required to compute a HOG descriptor (adapted from Dalal et al., 2006).

The first stage applies an optional global image normalization which is designed to reduce the influence of illumination effects. In practice, each pixel (r,g,b) and each color channel is normalized independently by computing the square root of its red, green, and blue color channels.

The second stage computes image gradients and orientations. This captures silhouette and texture information.

The third stage pools gradient orientation information by dividing the image into small spatial regions, called “cells”. For each cell a local 1-D histogram of gradient orientations, and of gradient magnitude, is built by accumulating the gradients of all the pixels in the cell.

The fourth stage takes local groups of cells and normalizes their associated orientation histograms. This step is introduced to achieve better invariance to illumination, shadowing, and edge contrast. Normalization is performed by measuring local histogram energy over groups of cells, referred to as “blocks”. This measure is then used to normalize the orientation histogram of each cell in the block. Typically, each individual cell is shared between several blocks, but its normalizations are block dependent and thus different. The cell thus appears several times in the final output vector with different normalizations. While this may seem redundant, it was shown quantitatively to improve performance (Dalal and Triggs, 2005).

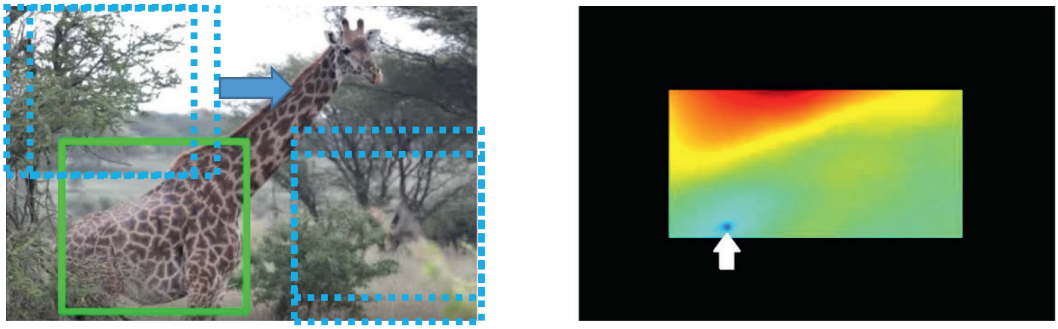

Fig. 3. Object detection using a sliding window approach. Input image (left), with a small subset of the sliding window positions illustrated by blue dotted rectangles, and the highest scoring window shown in green. Note that the classifier finds the giraffe torso accurately, even though the legs are cut off, and the back of the giraffe is occluded. Output of the Support Vector Machine classifier at each window position (right). Blue colors indicate locations where the classifier is most certain that the window contains the object (in this example, the torso of a giraffe). The white arrow highlights the center location of the window with highest detection score. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

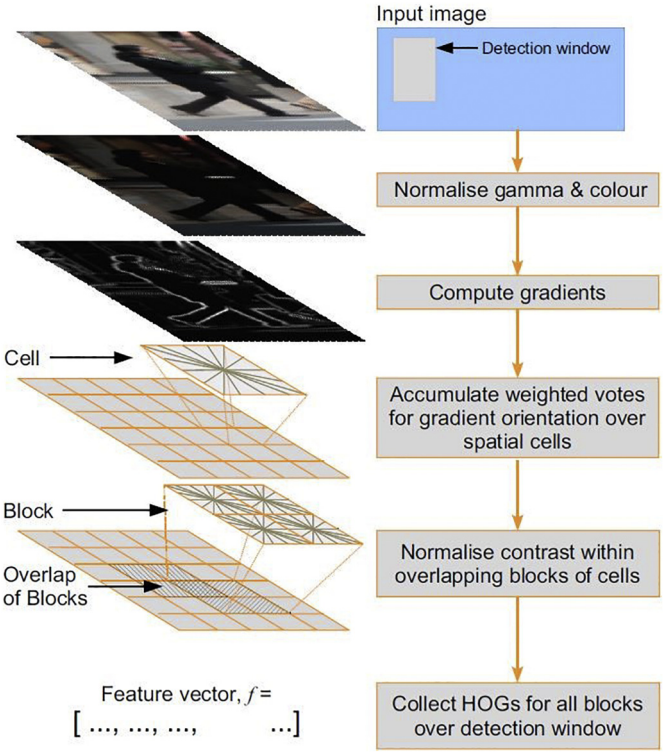

Fig. 4. Overview of HOG feature extraction. Given an input image or image patch, for each pixel the gradient magnitude and orientation is computed. The image is then divided into small spatial regions, called "cells". For each cell an orientation histogram is computed by accumulating the gradient orientation over all the pixels in the cell. These histograms are then further clustered into "blocks" and normalized to achieve better invariance to illumination, shadowing, and edge contrast. Finally, the HOG descriptor for the whole image is defined as the concatenation of all block responses.
Image adapted from Dalal et al. (2006).

The final step collects the histograms descriptors from all blocks of a dense overlapping grid of blocks.

This collection of histograms in our work consists of 4356 floating point values, and is used to train our object vs. background Support Vector Machine classifier (see Sections 2.1 and 2.2).

2.5. Implementation details

All our work is implemented in Python, and is deployed as a REST API to Azure using the Flask framework. We rely heavily on OpenCV, which is a popular library for real-time computer vision and contains the efficient implementation for HOG sliding window object detection introduced at the end of Section 2.4.

3. Data and results

3.1. Giraffe photographic capture-recapture study

We used data from a photographic capture-recapture study of individually identified, wild, free-ranging Masai giraffes in the Tarangire Ecosystem of Tanzania to train the automated cropping program, and to test its accuracy. Project GIRAFFE (http://www.wildnatureinstitute.org/giraffe.html) is a long-term, individual-based study examining giraffe demography (births, deaths, and movements) with the aim of estimating population size, reproduction, survival, and movements in an ecosystem with a range of anthropogenic effects on the landscape (Lee et al., 2016; Lee and Bolger, 2017).

We collected photographic data during daytime systematic road transect sampling. We sampled giraffes three times per year around 1 February, 1 June, and 1 October near the end of every precipitation season (short rains, long rains, and dry, respectively) by driving a network of fixed-route transects on single-lane dirt tracks in the study area. During sampling events, the entire study area was surveyed and a sample of individuals were encountered and approached so we could photograph the animal's right side at a perpendicular angle (Canon 40D and Rebel T2i cameras with Canon Ultrasonic IS 100–400 mm lens, Canon U.S.A., Inc., One Canon Park, Melville, New York, 11,747, USA). We attempted to photograph the right side of every giraffe encountered, and recorded sex and age class based on physical characteristics (Lee et al., 2016). We manually cropped all photos to include our area of interest, the torso from the lower neck to just below the belly or penile sheath and from chest to tail (Fig. 1). We collected 1800 photographs before and after cropping where the giraffe was in near-perfect profile to use as training data for the automatic cropping program.

3.2. Dataset

Our training dataset consisted of: (i) 500 annotated test images from February 2014, September 2014, and February 2015, where each image contained exactly one giraffe; (ii) 1300 annotated training images from other months; and (iii) 12,000 negative images which did not contain giraffes. These datasets were used to train the classifier, and to make parameter and design decisions.

3.3. Accuracy testing

We used the finished automated cropping program to crop 3518 raw

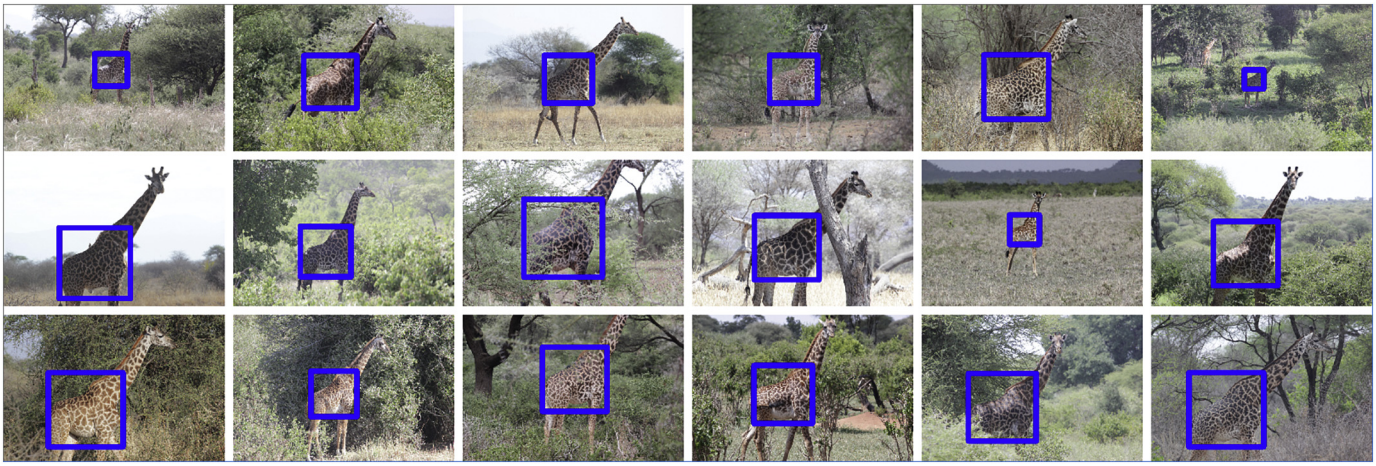

Fig. 5. Randomly selected giraffe torso detection results. Note that even giraffes which are small in the image (upper, left), or giraffes which are partly occluded by bushes, are found successfully.

data photos from three surveys in 2017 to assess the accuracy of the program. Raw data photos were unfiltered, and included high-quality images consisting of one animal in near perfect profile that nearly filled the frame, as well as low-quality images such as where the subject giraffe was very far away and thus very small, tilted in relation to the camera, and/or partially obscured by vegetation. Some images also included more than one giraffe, with the focal animal centered in the image. We ran the automated cropping program, then manually assessed the number of 'failures' where the cropping program did not accurately crop the complete torso area of interest of the focal animal.

Accuracy testing from 2017 surveys documented mean failure rate for all photos was 0.109, and mean failure rate for high-quality photos was 0.006. Most failures were extremely tilted and/or obscured by vegetation. High-quality photos of an unobscured giraffe in near perfect profile almost never failed to be cropped accurately by the program. We found that the system is very robust to scale changes, and can detect giraffes at very different positions and zooms (see Fig. 5, top right detection vs. the bottom right detection). Furthermore, the detector even finds the giraffe in environments where the giraffe seems to merge with the background.

All images used to train the giraffe detector are from 2014 and 2015 while the images for the testing were taken in 2017. Hence, the giraffe detection accuracy reported above reflects the true performance when applying the system to previously unseen images and in potentially new imaging conditions (e.g. different times of the day).

4. Discussion

Most photo-identification procedures require multiple processing steps, where all but the first step of selecting and cropping an area of interest to be matched have been automated to some degree. Our work here has demonstrated that a HOG descriptor and SVM model can efficiently identify and crop giraffe torso photos and remove a time-consuming step in processes aimed at fully automatic animal identification. The program developed here has been used successfully as part of an automated process to document wild, free-ranging giraffe demographic parameters that provide data-driven conservation recommendations for this vulnerable species (Lee et al., 2016; Lee and Bolger, 2017; Lee and Bond, 2016).

Using our HOG-detector we were able to obtain near perfect results. We not only found the object-of-interest, but only at a specific pose (side-facing) and did so with a tight detection rectangle. This is important, since the down-stream giraffe detection component relies on such tight and specific detections. HOG focuses on edges in the image, where each detection has to tightly match the expected HOG gradients. In comparison, Deep Learning represents a semantic classifier (especially in deeper layers), which is great for finding objects in all configurations and angles in an image, but less suitable for firing only on objects in a specific pose. Furthermore, Deep Learned classifiers tend to over-fire around an object, which is why non-maxima suppression gets performed in a post-processing step to merge (or discard) multiple detections into one. This step can introduce errors (e.g. pick less tightly fitting detections) and is of much less importance with HOG based detectors. To summarize, Deep Learning has been shown to work well on a wide range of scenarios, in part due to its ability to recognize an object independent of its pose, color, etc. In this work however, we were able to achieve near-perfect recognition results using a detection approach based on HOG descriptors. We can do so without expensive hardware requirements (a dedicated GPU). In addition, we only fire on the object-of-interest in an exact specified pose, and with a tight bounding box detection.

Computer vision-based automation is an essential process for turning digital images into useful data for ecological studies (Weinstein, 2017). The creation of images intended for ecological data analyses has outpaced the development of tools for automated processing (Swanson et al., 2015; Van Horn et al., 2017). We believe the automated cropping procedure we outlined here will greatly assist research programs seeking to turn images into data by removing one of the primary bottlenecks in the processing workflow. This program was developed and tested for use with giraffe images from a large, long-term demography study. Future work in this area should test this type of object detection for additional species and under different conditions, e.g. under dense forest canopy or under water.

Acknowledgments

We thank the Tanzanian Commission for Science and Technology and Tanzania Wildlife Research Institute for permission to conduct fieldwork. Financial support for this work was provided by Sacramento Zoo, USA, Columbus Zoo, USA, Cincinnati Zoo, USA, Safari West, USA, Tierpark Berlin, Germany and Tulsa Zoo, USA.

References

Adams, J.D., Speakman, T., Zolman, E., Schwacke, L.H., 2006. Automating image matching, cataloging, and analysis for photo-identification research. Aquat. Mamm. 32, 374–384. https://doi.org/10.1578/AM.32.2.2006.374.

Ardovini, A., Cinque, L., Sangineto, E., 2008. Identifying elephant photos by multi-curve matching. Pattern Recogn. 41, 1867–1877. https://doi.org/10.1016/j.patcog.2007.11.010.

Arzoumanian, Z., Holmberg, J., Norman, B., 2005. An astronomical pattern-matching algorithm for computer-aided identification of whale sharks *Rhincodon typus*. J. Appl.

Ecol. 42, 999–1011.
Bolger, D.T., Morrison, T.A., Vance, B., Lee, D., Farid, H., 2012. A computer-assisted system for photographic mark–recapture analysis. Methods Ecol. Evol. 3, 813–822.
Bradski, G., 2000. The OpenCV library. Dr. Dobbs J. 25, 120–126.
Burton, A.C., Neilson, E., Moreira, D., Ladle, A., Steenweg, R., Fisher, J.T., Bayne, E., Boutin, S., 2015. REVIEW: wildlife camera trapping: a review and recommendations for linking surveys to ecological processes. J. Appl. Ecol. 52, 675–685. https://doi.org/10.1111/1365-2664.12432.
Cuthill, I.C., 1991. Field experiments in animal behaviour: methods and ethics. Anim. Behav. 42, 1007–1014.
Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference. vol. 1. IEEE, pp. 886–893.
Dalal, N., Triggs, B., Schmid, C., 2006. Human detection using oriented histograms of flow and appearance. In: European Conference on Computer Vision. Springer, Berlin, Heidelberg, pp. 428–441.
Gamble, L., Ravela, S., McGarigal, K., 2008. Multi-scale features for identifying individuals in large biological databases: an application of pattern recognition technology to the marbled salamander *Ambystoma opacum*. J. Appl. Ecol. 45, 170–180.
Gope, C., Kehtarnavaz, N., Hillman, G., Würsig, B., 2005. An affine invariant curve matching method for photo-identification of marine mammals. Pattern Recogn. 38, 125–132. https://doi.org/10.1016/j.patcog.2004.06.005.
Kelly, M.J., 2001. Computer-aided photograph matching in studies using individual identification: an example from Serengeti cheetahs. J. Mammal. 82, 440–449.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc, Red Hook, New York, pp. 1097–1105.
Lebreton, J.-D., et al., 1992. Modeling survival and testing biological hypotheses using marked animals: a unified approach with case studies. Ecol. Monogr. 62, 67–118.
Lee, D.E., Bolger, D.T., 2017. Movements and source-sink dynamics among subpopulations of giraffe. Popul. Ecol. 59, 157–168. https://doi.org/10.1007/s10144-017-0580-7.
Lee, D.E., Bond, M.L., 2016. Precision, accuracy, and costs of survey methods for giraffe *Giraffa camelopardalis*. J. Mammal. 97, 940–948.
Lee, D.E., Bond, M.L., Kissui, B.M., Kiwango, Y.A., Bolger, D.T., 2016. Spatial variation in giraffe demography: a test of 2 paradigms. J. Mammal. 97, 1015–1025.
Lowe, D.G., 1999. Object recognition from local scale-invariant features. In: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. vol. 2. IEEE, pp. 1150–1157.
May, R., 2004. Ecology: ethics and amphibians. Nature 431, 403.
McCarthy, M.A., Parris, K.M., 2004. Clarifying the effect of toe clipping on frogs with Bayesian statistics. J. Appl. Ecol. 41, 780–786.
Minteer, B.A., Collins, J.P., 2005. Why we need an "ecological ethics". Front. Ecol. Environ. 3, 332–337.
Moya, Ó., Mansilla, P.L., Madrazo, S., Igual, J.M., Rotger, A., Romano, A., Tavecchia, G., 2015. APHIS: a new software for photo-matching in ecological studies. Ecol. Inform. 27, 64–70.
Petersen, S.L., Branch, G.M., Ainley, D.G., Boersma, P.D., Cooper, J., Woehler, E.J., 2005. Is flipper banding of penguins a problem? Mar. Ornithol. 33, 75–79.
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. Ecol. Inform. 36, 145–151.
Raj, A., Choudhary, P., Suman, P., 2015. Identification of tigers through their pugmark using pattern recognition. Open Int. J. Technol. Innov. Res. 15.
Rowcliffe, J.M., Carbone, C., 2008. Surveys using camera traps: are we looking to a brighter future? Anim. Conserv. 11, 185–186. https://doi.org/10.1111/j.1469-1795.2008.00180.x.
Sherley, R.B., Burghardt, T., Barham, P.J., Campbell, N., Cuthill, I.C., 2010. Spotting the difference: towards fully-automated population monitoring of African penguins *Spheniscus demersus*. Endanger. Species Res. 11, 101–111.
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. Sci. 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.
Van Horn, G., Mac Aodha, O., Song, Y., Shepard, A., Adam, H., Perona, P., Belongie, S., 2017. The iNaturalist Challenge 2017 Dataset. (arXiv preprint arXiv:1707.06642).
Van Tienhoven, A.M., Den Hartog, J.E., Reijns, R.A., Peddemors, V.M., 2007. A computer aided program for pattern-matching of natural marks on the spotted ragged tooth shark *Carcharias taurus*. J. Appl. Ecol. 44, 273–280.
Weinstein, B.G., 2015. MotionMeerkat: integrating motion video detection and ecological monitoring. Methods Ecol. Evol. 6, 357–362. https://doi.org/10.1111/2041-210X.12320.
Weinstein, B.G., 2017. A computer vision for animal ecology. J. Anim. Ecol. https://doi.org/10.1111/1365-2656.12780.
Williams, B.K., Conroy, M.J., Nichols, J.D., 2001. Analysis and Management of Animal Populations. Elsevier Academic Press, San Diego.
Wilson, R.P., McMahon, C.R., 2006. Measuring devices on wild animals: what constitutes acceptable practice? Front. Ecol. Environ. 4, 147–154.