ARTICLE
Emerging Technologies

An automated approach for counting groups of flying animals applied to one of the world’s largest bat colonies

Benjamin Koger1,2,3 | Edward Hurme2,3,4 | Blair R. Costelloe1,2,3 | M. Teague O’Mara2,4,5 | Martin Wikelski2,3,4 | Roland Kays6,7 | Dina K. N. Dechmann2,3,4

1Department of Collective Behaviour, Max Planck Institute of Animal Behavior, Radolfzell, Germany
2Centre for the Advanced Study of Collective Behaviour, University of Konstanz, Konstanz, Germany
3Department of Biology, University of Konstanz, Konstanz, Germany
4Department of Migration, Max Planck Institute of Animal Behavior, Radolfzell, Germany
5Department of Biological Sciences, Southeastern Louisiana University, Hammond, Louisiana, USA
6North Carolina Museum of Natural Sciences, Raleigh, North Carolina, USA
7Department of Forestry and Environmental Resources, North Carolina State University, Raleigh, North Carolina, USA

Correspondence
Benjamin Koger
Email: benkoger@gmail.com

Funding information
NVIDIA Corporation; Struktur- und Innovationsfonds für die Forschung of the State of Baden-Württemberg, the Deutsche Forschungsgemeinschaft, Grant/Award Number: EXC 2117-422037984

Handling Editor: Tanya Berger-Wolf

Abstract
Estimating animal populations is essential for conservation. Censusing large congregations is especially important since these are priorities for protection, but efficiently counting hundreds of thousands of moving animals remains a challenge. We developed a deep learning-based system using consumer cameras that not only counts but also records behavioral information for large numbers of flying animals in a range of lighting conditions including near darkness. We built a robust training set without human labeling by leveraging data augmentation and background subtraction. We demonstrate this approach by estimating the size of a straw-colored fruit bat (Eidolon helvum) colony in Kasanka National Park, Zambia with cameras encircling the colony to record evening emergence. Detection of bats was robust to deteriorating lighting conditions and changing backgrounds. Combined over five days, our population estimates ranged between 750,000 and 976,000 bats with a mean of 857,233. In addition to counts, we extracted wingbeat frequency, flight altitude, and local group polarity for 639,414 individuals. This open access method is an inexpensive but powerful approach that, in addition to radial emergences from central locations, can also be applied to unidirectional movements of flying groups, such as migratory streams of birds.

KEYWORDS
bats, computer vision, convolutional neural network, flight dynamics, image analysis, migration, population estimate
INTRODUCTION

Reliable estimates of population sizes are essential for conservation and understanding population dynamics (Marsh & Trenham, 2008; Turchin, 1999). This is particularly true for migratory species, which are difficult to follow over long periods of time, and face increasing threats and population declines across the globe (Harris et al., 2009; Wilcove & Wikelski, 2008). Repeated counts at stopovers or endpoints therefore provide crucial information about population sizes and trends.

Groups of flying animals can be particularly difficult to count as they typically roost in clumped aggregations that are disturbed by humans entering the colony. However, they are often easy to detect from the ground when they leave their roosts. Due to their large numbers and dispersed flight, counting them is challenging. Traditionally, such counts are conducted by eye, with observers either attempting to count every animal directly or counting subsets of animals and then extrapolating (Bildstein, 2004; Inuzzas et al., 2010; Robbins et al., 1989). Estimates based on live visual counts are subject to errors, especially at high densities (Forsyth et al., 2006; Westcott & McKeown, 2004). For example, colony-size estimates of Brazilian free-tailed bats (Tadarida brasiliensis) based on thermal images decreased compared with historical human counts by more than an order of magnitude, likely driven by humans overestimating group size for a small, fast-flying animal (Betke et al., 2008; Hristov et al., 2010). Thus, using automated approaches is essential for counting large groups of moving animals.

Recording flying animals with cameras, thermal detectors, or radar (Meade et al., 2019) can enable more precise and verifiable censuses, but counting the animals in the recordings is labor-intensive without computer automation. The best automated approach for counting animals in recordings depends on the characteristics of the footage, especially the level of contrast between the animals and their background and the amount of sensor noise. When contrast is high and sensor noise is low, relatively simple techniques, such as thresholding (where a brightness value defines the cutoff point between animals and the background) and background subtraction (where static background pixels are removed from all frames, leaving only the moving animals) are sufficient to efficiently and reliably detect all individuals (e.g., Betke et al., 2008). These conditions are achievable with visible light (RGB) cameras during daytime when dark animals are filmed against the bright sky. Footage quality deteriorates, however, in darker conditions. In low light conditions, heat-sensing cameras are often used to film animals (Betke et al., 2008; Cilulko et al., 2013; Hristov et al., 2008, 2010; Kays et al., 2018; McCarthy et al., 2021; Otlora-Ardila et al., 2020). This approach is effective, with reported human-validated errors of around 3% on average and maximum error <7% for automated flying bat counts (Betke et al., 2008; Otlora-Ardila et al., 2020). However, thermal cameras often cost many thousands of dollars (Otlora-Ardila et al., 2020) and their resolution is low compared with RGB cameras. Cost is therefore a challenge, especially if multiple devices are necessary to achieve sufficient coverage of the animal group.

Automated counting is challenging at dusk and dawn, when many animal groups are most active, not only because there is little natural light, but also because quickly changing environmental conditions require versatile detection models. Across a range of applications, convolutional neural networks (CNNs) have been effective at finding objects in complex natural scenes (Christin et al., 2019). CNNs are now commonly used for animal censuses and are robust in challenging real-world conditions (e.g., Guirado et al., 2019; Torney et al., 2019). However, these models generally must be trained on large human-annotated data sets. Creating these training sets is challenging, especially when a large number of animals appear in many different environmental conditions, and is compounded when the animals are hard for human annotators to see. To help overcome this challenge, a technique called data augmentation, where training images are modified to mimic a wider range of environmental conditions, is commonly used to increase the effective size of the training set with minimal extra effort (Shorten & Khoshgoftaar, 2019).

The primary objective of our study was to use the power of CNNs to enable the use of less expensive RGB cameras to detect and count animals in conditions where previously only thermal cameras could be used. Additionally, to ease the human burden of complex image annotation with many small animals, we investigated the ability of machine-annotated data paired with intense data augmentation to train a CNN without explicit human labeling.

Beyond population estimates, videos of flying animals also can contain information about flight behavior like average flight altitude, wingbeat frequency, local animal density, and movement polarization. While such fine-scale measurements are common in multi-camera 3D filming setups (e.g., Corcoran & Hedrick, 2019; Evangelista et al., 2017; Ling et al., 2018), those approaches are significantly more complicated to deploy and require at least three times the number of cameras. While calculating these values from single-camera videos has been discussed (Matzner et al., 2015) and at least partially implemented (wingbeat frequency in captivity, Atanbori et al., 2013; wingbeat frequency with posture matching, Breslav et al., 2012; and wingbeat frequency in birds, Li & Song, 2013), another specific objective of our study was to implement
this complete suite with single cameras and the output from the CNNs described above at scale in the wild.

We demonstrate that this method can reliably count and describe large numbers of individuals using videos of straw-colored fruit bats (Eidolon helvum) at one of the world’s largest bat colonies in Kasanka National Park (KNP) in Central Zambia as an example. The automated method for counting flying animal aggregations we present is observer-independent, repeatable, and widely deployable due to its use of imagery captured with inexpensive, consumer-level cameras.

MATERIALS AND METHODS

Overview

We studied a large transitory colony of E. helvum that roost in a swamp forest in KNP in Zambia. Every evening, almost the entire colony leaves to forage in the surrounding area (Calderón-Capote et al., 2020). Over five days, we used 10 small video cameras spaced evenly around the forest to film this emergence, starting before any bats were flying and ending after complete darkness. Using our software described below, every pixel of every frame of the footage was automatically labeled as either part of an object of interest (a bat) or as the background (Figure 1). From one frame to the next these objects were connected into movement trajectories, or tracks. Each track is a record of individual movement through the frame. To estimate total colony size, we counted the number of tracks that crossed the horizontal centerline of each video over the course of all observations, and, using the known relationship between the centerline and the geometry of the cameras paired with estimates of individual altitude, calculated the fraction of the total group filmed. We also quantified wingbeat frequency, animal height, and group polarization using further statistics derived from these tracks. We emphasize that other researchers using this method in other contexts should not necessarily use the exact parameter values we used for our example in KNP, but should instead choose parameters that match their system based on our explanations for how we chose our specific values.

Study site and video collection

Data were collected from November 16 to 20, 2019 at KNP in Central Zambia (12°13′00″ S, 30°11′40″ E). Up to several million migratory E. helvum roost here in a small area of mushitu “swamp” forest from October to January (Richter & Cumming, 2005, 2008; Sørensen & Halberg, 2001). Unlike bats roosting in a cave, bats here were able to leave the forest in any direction. Colony size typically peaks between mid to late November and early December and manual count estimates have ranged from 1 to 13 million bats (Richter & Cumming, 2005; Sørensen & Halberg, 2001; F. Willems, personal communication).

We surrounded the colony with 10 video cameras (GoPro, San Mateo, CA, USA; Models 4, 6, and 7, video settings: 25 frames/s, 2.7k pixel resolution) as evenly as possible (Figure 2). All cameras were set to linear mode in which the camera applies built-in post-processing lens distortion correction filters. We pointed the cameras directly up with the bottom of the frame pointing toward the roost. Cameras recorded from before evening emergence until complete darkness for five consecutive nights. On three of the nights, one camera shut off early because of lack of battery, and thus was excluded from those nights’ counts.

All code for subsequent video processing and analysis was written in Python programming language (Python Software Foundation) using the OpenCV (Bradski, 2000), numpy (Harris et al., 2020), and PyTorch (Paszke et al., 2019) packages. All code is available from Zenodo: https://doi.org/10.5281/zenodo.7955338.

Labeling object pixels

The process of labeling pixels in an image as one of a discrete set of classes is called semantic segmentation. Recently, techniques using CNNs have become effective for solving semantic segmentation problems (Long et al., 2015). We used a model based on a CNN called UNet that has been effective at a range of similar tasks (Ronneberger et al., 2015). For efficiency, instead of using the full 3D image (height by width by red green and blue color channels), we only input a 2D array (height by width recording intensity). Since most colors in our observation frames are gray and blue, to optimize contrast with dark bats, we used the pure blue channel as input. This model had to be trained with raw input images paired with the correctly labeled outputs. We used 4500 images for training and 898 for validation. The final performance of the counting method was tested as described in the Detection validation section. We used image augmentation as part of the model training pipeline to simulate a range of darkness conditions (see below and Appendix S1: Section S1 for training details).

Building the training set

Building an annotated training set to train a semantic segmentation model manually is time-consuming
because each pixel has to be individually labeled. Instead of manual annotation, we used background subtraction to label pixels in well-suited subsets of our videos, and then used image augmentation to simulate the conditions under which background subtraction fails. In our test case, these easier conditions occurred in the early evening when each bat contrasted strongly with the bright sky. To get as varied raw training data as possible, we made the automated annotation method itself relatively robust to dark conditions and camera sensor noise as explained below.

In our background subtraction annotation step, the image background was computed as the average of 31 frames centered on the frame being processed. The number of frames used for background subtraction was chosen based on qualitative performance checks on a handful of frames and was a trade-off between having enough frames that the moving bats were sufficiently
smoothed from the background while few enough that slower background movement from clouds did not cause false detections. Each pixel in the subtracted image with a value higher than a certain fraction (i.e., threshold) of the corresponding background pixel value was labeled as part of a possible object (bat) and all other pixels were labeled as background. We initially set a threshold of one tenth the background pixel value to distinguish bats from background; however, the specific value used in each frame varied with noise conditions. To detect the presence of many false detections driven by noisy pixels varying from the background by more than the threshold, we set an upper bound on the number of detections per frame where each detection was defined as a blob of pixels with the same label (see below). We used an upper bound of 600 objects per frame corresponding to roughly twice the observed maximum number of animals in a single frame based on humans viewing high-density video frames. We increased the detection threshold used for each frame until the number of detected objects fell below the upper bound.

All contiguous groups of pixels labeled as possible objects were grouped into larger object clusters (Figure 1). Only clusters greater than an area measured by a rotated minimal bounding rectangle were considered true objects; the rest were considered noise and dismissed. Each cluster had its own size threshold based on the local darkness of the background, which was related to expected sensor noise levels. The resulting clusters were then tracked from frame to frame using the tracking algorithm described in the Tracking section. Even in building the training set where we used nonconsecutive frames, tracking is important because most sensor noise is ephemeral and, compared with real objects, will not be tracked. Objects with a track length of one were considered noise and excluded. For a given training frame, the object pixel labels come from all tracked clusters that are present in that frame.

Over the course of each evening observation, darkness, and therefore camera sensor noise, increases. Visual inspection was used to decide when mislabeling became too extreme to be useful for each video used to build the training set. Since UNet performance has been shown to be robust even in the presence of slightly noisy annotations (<10%) (Dineanu et al., 2022), visual inspection is efficient because we can ensure the majority of
annotations are correct in a given frame in a few seconds instead of many minutes or more per frame with explicit human labeling. Within this time range, images were randomly selected for our training and validation sets.

The resulting training set contains only images that are relatively bright, with high contrast and little noise, not all observed conditions. However, across an evening only the lighting and noise on the images changed, not the physical size and shape of the bats; therefore, we simulated more challenging conditions with image augmentation. Specifically, we altered the training images by decreasing brightness and contrast, and adding Gaussian blur and Poisson noise (see Appendix S1: Section S1 for the specific augmentation regime).

**Tracking**

We used the Hungarian algorithm (Kuhn, 1955; as implemented by SciPy, Virtanen et al., 2020) with additional rules to connect detections between frames. This algorithm minimizes the total distance between all pairs of connected points in a pair of frames. We minimize over the natural log of the distance between pairs so that long-distance false pairs do not dominate the overall optimization. If an existing track is farther than a maximum distance threshold (minimum of either 30 pixels or 0.45 times the distance to the nearest neighbor) from any new point, we assume there was a detection error and estimate that track’s position in the next frame as the last detection location. The maximum distance threshold is chosen to be above but near the actual step size of the observed animals to prevent adding points to a track that comes from other bats or noise. We used a value, 30, three times the average step size for tracks in our data and confirmed afterward that 99.96% of measured steps were below 29 pixels. The alternative threshold of 0.45 multiplied by the distance to the nearest neighbor creates a more conservative threshold in higher density areas to minimize tracks switching between different individuals. If a new point is not added to a track for two consecutive frames, it is considered finished. If a new point is connected to a track within two frames, then any missing locations are linearly interpolated. All new points not added to existing tracks are used to start new tracks; however, if they do not pair with a new point in the next frame, they are considered noise and deleted.

**Counting animals on video**

To count individuals, we counted all tracks that cross the horizontal frame midline and recorded the direction of travel across the line. Animals traveling up (away from the colony) were added, and animals traveling down (toward the colony) were subtracted from this sum. We used human validation to estimate the average accuracy of our counts as a function of each frame’s average darkness (see Detection validation) since we expected the model to have more trouble in the darkest conditions. To account for this bias, we then multiplied the bat count in each frame by the inverse of the human-measured accuracy corresponding to that frame’s darkness level.

**Extrapolating to total population counts**

When all individuals in an aggregation are filmed, the corrected count from the video footage is sufficient to estimate the total aggregation size. However, if only a subset was filmed, it is necessary to determine what fraction of the population was counted so the count can be extrapolated.

In our example, almost all bats foraged well outside the roost. If we could record all bats crossing an imaginary cylinder around the roost, we would record the entire population. Since our 10 cameras did not cover this entire area, we instead calculated what fraction of this imaginary cylinder our cameras monitored and then extrapolated the total population.

To estimate the proportion of the bat colony we counted, we first calculated, based on each camera’s field of view, the length in meters of the horizontal midline across each video frame as a function of bat height (Figure 3). Since this countline gets wider farther away from the camera, the proportion of the population we recorded crossing this line at each camera depended on the flight altitude of each bat, with a greater proportion of high-flying bats in the total population captured compared with low-flying bats. We estimated the flight altitude of each bat by measuring its wingspan in pixels while assuming each bat’s actual size was equal to the species average of 0.8 m (Bergmans, 1988; DeFrees & Wilson, 1988; Riskin et al., 2010). To estimate the wingspan in pixels, we measured the largest side of the minimum bounding rectangle for each bat blob in every frame in every track and then averaged the sizes of the peaks in this measurement (see Appendix S1: Section S2.3). This assumes that bats were generally at a fixed height when crossing the frame, which matched our observations each evening while filming.

The distance of each camera from the bat forest also affected the proportion of the aggregation it captured: cameras placed closer to the colony covered a greater fraction of the imaginary count cylinder surrounding the colony (Figure 3), and therefore a larger proportion of the population. We assumed that all individuals emerged from a single point, and calculated the distance of each camera from this point.
Taking into account the camera location, field of view, and each recorded bat’s flight elevation, we multiplied the count of bats at each altitude by the inverse fraction of the environment captured at that altitude, or the “count multiplier” (see Appendix S1: Section S2).

The spatial distribution of the bats’ flight paths also affected how we scaled from single-camera counts to population estimates: if individuals flew in all directions with equal probability, we could have simply extrapolated from a single camera’s count or from the average of multiple cameras’ counts. In our system, however, possibly because of factors like weather and food availability, different densities of bats flew out in different directions. We therefore estimated the density of bats leaving at every degree around the roost from the densities recorded at each pair of neighboring cameras. As described in more detail below, we transitioned from the density measured at one camera to the density measured at the second camera somewhere midway between the two. The total population was estimated by summing these densities around the entire circle.

The final population estimate across multiple cameras was then a function of both the estimated location of the colony’s center and where the local bat density transitioned from one camera to the next. Neither the exact center nor the exact transition points were known. Instead, for each day, we estimated the population 1000 times using a randomly chosen forest center and randomly drawn camera density transition points for each estimate (see Figure 2 for visualization and Appendix S1: Section S2.5). To estimate the number of bats present over the course of the five-day period, we randomly drew one daily population estimate from each day’s distribution (of 1000 estimates) and averaged the resulting five values. We repeated this process 1000 times (with replacement), resulting in a distribution of average population estimates from across the five days and the range of possible variation in our estimate.

**Calculation of additional behavioral information**

As part of our counting procedure, we tracked the movement, estimated the altitude, and measured the body profile of every detected individual. These measures offer behavioral insights with biological implications beyond calculating population size.

We measured the distribution of departure directions for the colony as a whole each night by combining the measures of bat density from different cameras. This was done per degree around the estimated colony center point to account for the fact that cameras were at variable distances from the roosting site.

We measured the wingbeat frequency based on the change in size of each tracked bat across frames (Figure 1). This approach was feasible because the bats were engaged in long-distance flight rather than, for example, rapid turning or hunting maneuvers that would generate variation in their orientation relative to the
camera’s image plane. Each track’s size measurements were noisy and relatively brief (a few seconds). Therefore, we used Welch’s method with a Hann window as implemented in SciPy to estimate the time averaged power spectrum for each track’s wing dynamics (Virtanen et al., 2020; Welch, 1967). We used the dominant nonzero frequency as the bat’s wingbeat frequency. We only examined tracks greater than 100 frames (4 s).

From the movement trajectories, following Couzin et al. (2002), we also calculated overall group polarity for each camera (Equation 1). 

\[ p_{\text{group}}(t) = \frac{1}{N} \left| \sum_{i=1}^{N} v_i(t) \right|, \]

where \( p_{\text{group}} \) is group polarization and \( v_i \) is the movement unit vector of each individual and \( N \) is the total number of individuals. To quantify global movement, we constructed each track’s movement unit vector by normalizing the vector pointing from the first point to the last point in each track. Here we also only examined tracks of length 100 frames and up.

**Detection validation**

We used human validation to assess the performance of our bat detection pipeline. To do so, we watched video segments in which our software highlighted all tracked bats as they crossed the frame midpoint. The observer then counted the number of bats crossing the midline that were not detected and cases where a bat was detected but not actually present and the direction of this false detection (up or down across the midline).

We first extracted five 15-s clips from each camera, spacing the clips evenly over the first two days of each camera’s footage. We found that bat detections decreased at the end of each day’s filming; however, the temporal resolution of our validation sampling was insufficient to distinguish between an actual decrease in animals and a decrease in software performance. Therefore, to investigate the end of observations, we extracted a second set of clips consisting of the 7 s after 65%, 80%, 95%, and 98% of detected bats had passed for each camera over the first two days. We found that by rescaling the brightness of the video, it was possible for a human validator to detect most bats present throughout the observation, even in very dark frames.

We expected count accuracy to decline with decreasing brightness. The results from both sets of validation clips showed relatively constant accuracy across observations until a certain average frame darkness, at which point accuracy deteriorated. We fit a piecewise linear function with each accuracy estimate weighted by the number of bats in that clip. These values were consistent across a range of initial parameters. We used this function to scale the number of bats counted in each frame based on the frame’s average darkness.

**Wingspan validation**

To estimate the height of each bat, we combined an estimate of the bats’ wingspans with information about the geometry of the camera being used. To validate how accurately our automated process measured the wingspan, we randomly chose 100 tracks from each camera from the first two nights. For each track, we randomly chose a frame in which the bat’s wings were measured as fully outstretched. Then, using a custom-built graphical user interface (GUI), a human clicked on each wingtip in a close-up image of the bat (see Appendix S1: Figure S3 for visualization of the GUI). We compared the resulting groundtruth distance between wingtips with the distance measured by our software and calculated a scaling term to correct the software measurement. We verified that the measured trends hold across days and camera groupings (see Appendix S1: Section S4). As the average frame brightness decreased, the mean and variance of the distribution of scaling values also changed. Therefore, we split the average frame brightness range measured across our validation tracks into four evenly sized bins and then used kernel density estimation to approximate the distribution of the correction terms as a function of body size. We used a Gaussian kernel with a SD that was half the SD of the data in each bin.

In practice, for each crossing bat, we scaled the bat’s estimated wingspan by randomly pulling a correction value from the distribution corresponding to the brightness of the frame when the track crossed the frame midpoint.

**RESULTS**

To test the performance of this method, we recorded the evening departure of straw-colored fruit bats, *E. helvum*, from a large colony in KNP, Zambia over five nights.

**Estimating population size**

For each of five filming days, we calculated 1000 daily bat population estimates using randomly chosen forest centers and camera region transition points. The lowest
daily average population estimate was 728,831 bats on November 20. The highest daily average population estimate was 987,114 bats on November 18. The lowest single estimate was 544,564 bats on November 17, while the highest was 1,297,738 bats on November 18 (Figure 4). The averaged estimates sampled over the five days ranged from 754,759 to 988,806 bats. The mean population estimate over the five-day period was 857,932 bats (95% CI [−3213, +1560]) (Figure 4).

Behavioral measurements

We found large variation in the density of bats that emerged around the roost with a strong bias toward the southeast (Figure 5). The highest recorded density of bats over an evening was 9999 bats per degree recorded on November 18 at the FibwePublic camera. The lowest density of bats measured was 175 bats per degree recorded on November 16 at the Puku camera. This large variation makes clear that population estimates calculated from a single location will generally vastly under- or overestimate the total population. Total population estimates using data from just a single camera range from 36,000 to 3,732,964 bats (see Appendix S1: Section S5).

Over the five days of filming, we recorded 710,409 tracks that were 4 s in length or greater. For each track, our model calculated the bat’s average height and dominant wingbeat frequency (see Figure 5 for example flight statistics from November 16 and Appendix S1: Section S6 for all day’s distributions). We validate the accuracy of these values in the following subsections.

Detection validation

Across the evenly spaced validation clips, we saw an overall bat count accuracy of 94%. Across all clips that contained at least one bat, the weighted median count accuracy was 95.3%, with a lower quartile accuracy of 92.0% and an upper quartile accuracy of 97.5%. Across the 7-s clips covering the darkest part of our videos, we achieved an overall accuracy of 88%. This compared to an overall accuracy of 70% on the same 7 s validation clips for the non-deep learning-based model, which was the actual model that annotated our training data.

Across most light levels, the accuracy remained relatively consistent, but then began to sharply decrease (Figure 6a). We fit a weighted two-piece piecewise linear function weighted by the number of bats in each validation clip. The SE of residuals was 0.049 for the resulting model (function 1: \( y = 0.072x + 0.195 \); function 2: \( y = -0.00012x + 0.939 \); breakpoint: 16, where \( x \) is the average brightness of the frame and \( y \) is the accuracy of the count in that frame). This function was used to scale the count across all observations. We also analyzed the performance of the two validation nights separately to confirm these results were consistent across days (see Appendix S1: Section S7).

The majority of bats crossed while brightness levels were high enough to enable accurate detection (Figure 6c). For all cameras, the human-verified count of bats decreased during the same period we saw a drop-off in the number of bats our software counted (see Appendix S1: Section S7). We also see that at the darkest levels, there was an increase in short tracks (<5 frames) that were likely mostly noise (Figure 6c). For counting animals, most noise canceled out (see Appendix S1: Section S8); however, this indicated that for investigating track-level attributes like wingbeat frequency and height, it made sense to remove shorter tracks.

Wingsize validation

Using human validation, we calculated the difference between the wingspan in pixels measured by our software and the true wingspan in pixels as measured by a human, and recorded this error normalized by the software-measured wingspan. We split up the validation wingspans into four evenly spaced bins based on the average brightness of the frames the wingspans came from. On average, our software, without correction,
FIGURE 5  Legend on next page.
overestimated the wingspan of the bats and did so to a greater extent with increased darkness (Figure 6b).

Correcting for this error, we investigated the resulting precision of the count multiplier and height across measured wingspans. With a camera 400 m from the forest center and average brightness in the range of the third bin, across all tracks, the 2.5th percentile wingspan (7 pixels) had a median count multiplier of 7.5 (quartiles: 6.8, 8.1). The corresponding altitude had a median value of 190 m (quartiles: 176.0, 211.3). The median wingspan (24 pixels) had a median count multiplier of 24.6 (quartiles: 22.1, 26.5) and a corresponding median altitude of 55.3 m (quartiles: 51.3, 61.6). The 97.5th percentile wingspan (48 pixels) had a median count multiplier of 48.9 (quartiles: 44.0, 52.7) and corresponding median altitude estimate of 27.7 m (quartiles: 25.7, 30.8). See Appendix S1: Section S9 for full distributions.

DISCUSSION

We showed that even with standard visible light consumer cameras, it is possible to build a system that monitors huge numbers of flying individuals in low light conditions. The combination of simple computer vision techniques to create labels that then function to train a deep learning-based detection model fuses the benefits of both approaches. It removes the need for explicit human annotation while creating a system that, compared with traditional thresholding techniques, is more robust to variable conditions, allowing us to provide a robust reproducible population estimate for the largest African bat colony. In addition to monitoring population size, this system also allows us to examine finer scale behavioral information for a number of animals currently unachievable with human-based or tagging-based techniques.

Our system builds on well-tested methods that use computer vision to automatically count bats from thermal sensors as they fly out of a cave roost (Betke et al., 2008; Hristov et al., 2010; Stepanian & Wainwright, 2018). Our system works with inexpensive RGB cameras, eliminating the need for more expensive thermal cameras, and also works over longer distances because of the better resolution of RGB compared with thermal in general. While our method does not work in complete darkness, it works well down to low levels of twilight. The ability to work in the dark will improve as commercial RGB cameras improve their low-light sensitivity, as some of the cameras we used were already five years old.

While deep learning is used more and more to detect animals in the wild, so far background subtraction and thresholding-based methods remain the standard technique for flying animal detection when filmed from below. This is likely because heat-sensing cameras and RGB cameras used during the day filming against a bright sky generally produce clean high contrast images, making the added work of annotating data outweigh potential gains. However, the necessity of using heat-sensing cameras or recording in well-lit environments excludes many use cases, including the one demonstrated in this paper, highlighting the utility of more advanced algorithms like the one we proposed here.

Using consumer RGB cameras and automated software to film and count bats enabled us to monitor a colony from multiple locations simultaneously and ultimately combine each camera’s recording into a total population estimate. While this adds complexity, we provide an approach that propagates error along each step and thus provides a robust estimate and CI. This need for many cameras is emphasized by the two order of magnitude difference we see in population estimates when using data from only a single camera. This may also explain the discrepancy from human visual counts if they were made at a position where the flyout was concentrated. This echoes the study by Westcott et al. (2012), which argues that, at larger scales, the most important parameter for monitoring overall population dynamics is not actually the accuracy of any given count but instead the fraction of the population that is actually included in a survey. Data from multiple cameras also allow researchers to more explicitly determine the appropriate number of cameras for a given survey. While ideally one might have enough cameras for complete coverage, in many contexts, including ours, that would be both logistically and financially infeasible. Instead, as we show, one can simulate counts with data from randomly chosen subset of cameras to confirm enough cameras have been deployed to generate counts that are robust to specific number and placement of cameras. RGB cameras and well-trained software make this type of widespread monitoring more feasible.

In addition to heat-sensing cameras, the other traditional method for flying animal censuses is human

**FIGURE 5** Example flight statistics from November 16 extracted by our software. When relevant, we use the same estimated forest center location as Figure 2. For reference, the distance from each camera to the estimated forest center is shown in the gray plot at the top of the figure. (a) Bats per degree over whole night as measured by each camera. (b) Polarization of all bats across the entire night for each camera. (c) Distribution of middle 90% of estimated wingbeat rates from all 105,717 recorded tracks on the 16th of length 4 s or above. (d) Middle 90% of all heights measured from each camera from tracks recorded for 4 s and longer.
counts (Hayman et al., 2012). While probably appropriate for smaller colonies where each animal can literally be counted, larger colonies pose problems. Both Westcott and McKeown (2004) and Forsyth et al. (2006) ran experiments to validate the accuracy of human counts of Australian flying fox species. They each found that density has a large effect on both the bias in human counts and their variability, and that the error becomes problematic at densities well below what we encounter at KNP. This finding is probably reflected in the past human counts from our site over the last 20 years, which range from 1.5 million bats in 1996 (Sørensen & Halberg, 2001) to a maximum of 10 million bats around 2005 (Richter & Cumming, 2005). While it is possible, based on reported counts, that the bat population experienced a rapid increase from 1996 to 2005 and then a rapid decrease from 2005 up to our count in 2019, when we use only the data from our camera placed near where Sørensen and Halberg made their count, we produce population estimates generally similar to their estimate because that is a high bat density part of the flyout (see Appendix S1: Section S10 for a more detailed investigation). This indicates the difference in estimates between Sørensen and Halberg, and ours is driven by a change in method and not the underlying population. Unfortunately, Richter and Cumming give no information about the origin of their estimate from 2005. However, park staff who have worked continuously over this period have not noticed an obvious increase or decline in numbers (Kennedy, personal communication). Taken together, similar to the adjustment to population estimates of the Brazilian free-tailed bat caves of the southwestern United States after automated counts were used (Betke et al., 2008), we suggest that our present estimate of around 850,000 bats for the days we recorded in 2019 differs from prior estimates largely because of previous methodological limitations. Ultimately, though, because of the range of methods deployed, the available evidence does not allow us to rigorously estimate how the true bat population may have changed over the previous two decades, and we therefore believe our count should be considered separately from historical counts when considering things like population trends. Importantly for future monitoring, since our method proceeds algorithmically from raw video all the way to final count, we explicitly defined in software every assumption made to reach our final estimated population distributions. While we made assumptions that best matched our experience with the system and our validation data, our method allows us and others in the future to explicitly model how other sets of assumptions and uncertainties affect the final outcome. This type of investigation can highlight which assumptions of the method most drive uncertainty to guide additional accuracy in future counts.

**FIGURE 6** Method accuracy in relation to brightness of the sky. Higher numbers on the x-axis correspond to brighter conditions on a scale of 0–255. (a) Brightness versus detection accuracy. Each circle represents a different human validation clip where circle size corresponds to the total number of bats in the clip and the circle’s center is the human-validated accuracy for that clip. Darker regions result from overlapping circles. The gray solid line is the best fit for a piecewise linear function weighted by bats per second in clip. (b) The kernel density estimated distributions of wing correction factors. The gray dashed line corresponds to no correction. Ten outliers <−1 are not shown. (c) Distribution of tracks across darkness conditions. The vertical gray dashed lines are the bin boundaries in (b). Tracks of length five frames or fewer are highlighted in light gray.
While our system was built initially to deal with large bats, there are many large flying animal aggregations that would benefit from efficient and robust recording solutions including seabird colonies and diurnal bird migration flyways (e.g., Gibraltar, Martin et al., 2016, Veracruz, Inzuzas et al., 2010). See Appendix S1: Section S11 for suggestions on applying this method to other systems. One limitation of our system is that we assume the size of our animals to estimate their heights. This could be particularly problematic for situations where there are many species flying together. For a more careful approach, similar to Atanbori et al. (2018) and Li and Song (2013), after extracting individual tracks, one could use an additional specifically trained model to extract species identity and then use appropriate size estimates for each species.

Our method also provides flyout statistics, like wingbeat frequency and group polarization, at an unprecedented scale in terms of animals surveyed. This provides a new window into how animals fine-tune behavior in different conditions. Riskin et al. (2010) report that, in lab conditions, large bats’ speed is inversely proportional with wingbeat frequency. O’Mara et al. (2019) found wingbeat frequency increases slightly with tailwinds in *E. helvum*, and the strength of those wingbeats decreases with tailwind support. Wingbeat, therefore, begins to give us a window into the behavior and environment of the bats. Currently, three papers report a combined 64 wingbeat measurements of our study species from a combination of accelerometry tags and video with wing beat distributions of 4.4 ± 0.43 beats/s \( (n = 7) \) (Norberg & Norberg, 2012), 4.5–5.7 beats/s \( (n = 10) \) (Carpenter, 1986), and 4.07 ± 0.28 beats/s; minimum: 3.76 and maximum: 5.96 \( (n = 47) \) (O’Mara et al. 2019). While our measured distributions overlap with these measurements to some extent, our values, measured from 639,414 individuals over five days, are generally lower. This is potentially consistent with the fact that, unlike some of the other studies where bats were recorded near takeoff, we film the bats during a long-distance directed movement when we might expect them to be flying consistently fast and therefore with lower wingbeat frequency. While outside the scope of this paper, it is exciting to note the level of both between camera and between day variation in frequency distributions indicating room for exciting future possible analysis.

Although bats flew out of the colony in all directions, there was a consistent preference toward the southeast during the five days we monitored the colony. Given that all 20 bats tracked by Calderon-Capote (Calderon-Capote et al., 2020) on their nightly foraging visits flew straight lines from the roost to a feeding tree, it is reasonable to extend our departure directions to infer colony-wide feeding directions. Additionally, combining this insight with each camera’s recorded group polarization gives insight into how ubiquitous a specific flight direction is at each measured location. Monitoring this over more of the season when bats are present, and across years, could help confirm where the most important feeding areas are for this bat colony.

**CONCLUSION**

We present a method to reproducibly monitor a large and dynamic colony of flying animals in challenging conditions. By using inexpensive cameras paired with robust software, we are able to monitor the colony from multiple locations and show that, in fact, because of large variation in animal density, no single location is sufficient for robust monitoring. Although our overall population estimate falls below previously reported values, when we account for the documented assumptions of previous methods, our estimates are actually within the error range of prior counts. This highlights the necessity of explicit modeling assumptions for long-term population monitoring. Ultimately, the combination of reproducible and interrogable population estimates paired with individual behavioral information opens the door to richer monitoring of important yet hard to study animal populations.

**AUTHOR CONTRIBUTIONS**

All authors conceived of the project. Benjamin Koger, Blair R. Costelloe, M. Teague O’Mara, Martin Wikelski, Roland Kays, and Dina K. N. Dechmann collected the data. Benjamin Koger designed and implemented the method. Benjamin Koger led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

**ACKNOWLEDGMENTS**

We would like to thank Kasanka National Park and in particular Geraldine Taylor and Richard Peel. We would also like to thank Frank Willems for providing count information, Alex Jordan and Iain Couzin for providing cameras, the Struktur-und Innovationsfonds für die Forschung of the State of Baden-Württemberg, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy—EXC 2117-422037984, and the NVIDIA Corporation for the donation of a TITAN Xp GPU. Open Access funding enabled and organized by Projekt DEAL.

**CONFLICT OF INTEREST STATEMENT**

The authors declare no conflicts of interest.

**DATA AVAILABILITY STATEMENT**

Code (Koger et al., 2023a) is available from Zenodo: https://doi.org/10.5281/zenodo.7955338. Data (Koger et al., 2023b)
there are available from the Edmond Repository: https://doi.org/10.17617/3.1YIESH.

ORCID
Benjamin Koger  https://orcid.org/0000-0001-7092-5314
Edward Hurme  https://orcid.org/0000-0002-2936-2449
Blair R. Costelloe  https://orcid.org/0000-0001-5291-788X
M. Teague O’Mara  https://orcid.org/0000-0002-6951-1648
Roland Kays  https://orcid.org/0000-0002-2947-6665
Dina K. N. Dechmann  https://orcid.org/0000-0003-0043-8267

REFERENCES


**SUPPORTING INFORMATION**

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