



OPEN ACCESS

EDITED BY

Craig Perl,
Arizona State University, United States

REVIEWED BY

Cody A Freas,
University of Alberta, Canada
Midori Sakura,
Kobe University, Japan

*CORRESPONDENCE

Jesse Rudolf Amenuvegbe Wallace

✉ jesse.wallace@csiro.au

Eric Warrant

✉ eric.warrant@biol.lu.se

RECEIVED 14 June 2023

ACCEPTED 21 August 2023

PUBLISHED 13 September 2023

CITATION

Wallace JRA, Reber TMJ, Dreyer D,
Beaton B, Zeil J and Warrant E (2023)
Camera-based automated monitoring of
flying insects (Camfi). I. Field and
computational methods.

Front. Insect Sci. 3:1240400.

doi: 10.3389/finsc.2023.1240400

COPYRIGHT

© 2023 Wallace, Reber, Dreyer, Beaton, Zeil
and Warrant. This is an open-access article
distributed under the terms of the [Creative
Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The
use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

Camera-based automated monitoring of flying insects (Camfi). I. Field and computational methods

Jesse Rudolf Amenuvegbe Wallace^{1,2*},
Therese Maria Joanna Reber³, David Dreyer³, Brendan Beaton¹,
Jochen Zeil¹ and Eric Warrant^{1,3*}

¹Research School of Biology, The Australian National University, Canberra, ACT, Australia, ²National Collections & Marine Infrastructure, CSIRO, Parkville, VIC, Australia, ³Lund Vision Group, Department of Biology, Lund University, Lund, Sweden

The ability to measure flying insect activity and abundance is important for ecologists, conservationists and agronomists alike. However, existing methods are laborious and produce data with low temporal resolution (e.g. trapping and direct observation), or are expensive, technically complex, and require vehicle access to field sites (e.g. radar and lidar entomology). We propose a method called “Camfi” for long-term non-invasive population monitoring and high-throughput behavioural observation of low-flying insects using images and videos obtained from wildlife cameras, which are inexpensive and simple to operate. To facilitate very large monitoring programs, we have developed and implemented a tool for automatic detection and annotation of flying insect targets in still images or video clips based on the popular Mask R-CNN framework. This tool can be trained to detect and annotate insects in a few hours, taking advantage of transfer learning. Our method will prove invaluable for ongoing efforts to understand the behaviour and ecology of declining insect populations and could also be applied to agronomy. The method is particularly suited to studies of low-flying insects in remote areas, and is suitable for very large-scale monitoring programs, or programs with relatively low budgets.

KEYWORDS

Camfi, population monitoring, flight behaviour, insect conservation, insect ecology, remote sensing, computer vision, image analysis

1 Introduction

The ability to measure flying insect activity and abundance is important for ecologists, conservationists and agronomists alike. Traditionally, this is done using tedious and invasive methods including nets (e.g. 1), window traps (e.g. 2), light traps (e.g. 3, 4), and pheromone traps (e.g. 5, 6), with the latter being favoured by agronomists for its

specificity. The WWII development of radar led to the introduction of radar ornithology (7, 8) and ultimately radar entomology (9, 10), which facilitated non-invasive remote sensing of insects flying up to a couple of kilometres above the ground, and became extremely important for understanding the scale and dynamics of insect migration (11). More recently, entomological lidar has been introduced, which benefits from a number of advantages over radar, in particular the ability to measure insects flying close to the ground, without suffering from ground clutter (12, 13). However, both entomological radar and entomological lidar systems are relatively large (requiring vehicle access to study sites), bespoke, expensive, and require expertise to operate, reducing their utility and accessibility to field biologists.

We propose a method for long-term population monitoring and behavioural observation of low-flying wild insects using wildlife cameras. The proposed method, described herein, combines simple and inexpensive field techniques with an advanced, open-source, and highly automated computational processing workflow based on Mask R-CNN (14). The method therefore lends itself to large-scale studies and can generate substantial volumes of behavioural and abundance data, allowing for the detection of subtle interactions between external factors, and insect abundance and behaviour.

The method permits study designs in which cameras are deployed at fixed points—potentially over long durations—which are used to detect insects flying past the camera. It is therefore best suited to studies which can make use of detection count and flight telemetry data, but is not suitable for analysing inter-individual interactions or individual trajectories over a larger area. Depending on the research question, the method can make use of either still images or video clips. The former enables long-term population monitoring in remote areas with only occasional visits by the field worker and produces (relatively) compact datasets which are amenable to manual or automatic image-annotation, while the latter enables rapid measurement of oriented flight behaviour from large numbers of insects over short periods of time. Under certain circumstances, the method also facilitates crude measurements of wingbeat frequency, allowing the researcher to exclude non-target species with very different wingbeat frequencies from their analyses.

In this paper, we describe our new method, including algorithms for automatic annotation of images of flying insects and tracking multiple insects in video clips. We also present a command-line program and python library, called Camfi, which implements the described procedures and characterises the performance of the automated annotation algorithm. In the preceding paper in this journal, we demonstrate the utility of the new method by measuring activity levels and flight behaviour of migratory Bogong moths (*Agrotis infusa*) over two summers in the Australian Alps (15). The Bogong moth is an important source of energy and nutrients in the fragile Australian alpine ecosystem (16), and is a model species for studying directed nocturnal insect migration and navigation (17–19). A dramatic drop in the population of Bogong moths has been observed in recent years (20, 21), adding it to the growing list of known invertebrate species whose populations are declining (22). The present method will prove invaluable for ongoing efforts to understand the behaviour

and ecology of the Bogong moth, and to monitor the population of this iconic species. The new method allows for straight-forward training on new datasets and other flying insect species, giving it promising applications within insect conservation and agronomy beyond the Bogong moth study system to which it has currently been applied.

2 Methods

Where indicated, the methods described below have been automated in our freely available software and python library, called Camfi. A full practical step-by-step guide for using these methods along with complete documentation of the latest version of the code is provided at <https://camfi.readthedocs.io/>.

2.1 Data collection

Images and video clips are collected in the field using wildlife cameras equipped with an infra-red LED flash and a “time-lapse mode” (we used BlazeVideo, model SL112, although most wildlife cameras or other infra-red cameras may be suitable). The cameras are positioned pointing towards the sky or other plain background (see sample placement in Figure 1), and set to take photos or short (e.g. 5 s at 30 frames per second, 1080p) video clips at regular intervals during the night (the PIR motion sensors on wildlife cameras generally do not trigger captures for invertebrates, as they depend on body heat, which limits them to detecting endotherms). The cameras are deployed in locations known for an abundance of flying insects of a known target species. When deployed in areas with a mixture of species, light traps can be deployed nearby to characterise the composition of the insect population in the area.

To train the automatic annotation model described below, we collected images from various locations known to be abundant in Bogong moths in the alpine areas of south-eastern Australia. These data are presented in further detail by Wallace et al. (15).

2.2 Image and video annotation

Our method permits three approaches to annotating the images and/or videos captured by the cameras in the field: 1. Manual annotation of still images using VIA (version 2; 23), 2. Automatic annotation of still images using Camfi, with optional validation or editing using VIA, and 3. automatic annotation of video clips using Camfi. When the target insects are flying sufficiently fast and the exposure time of the cameras is sufficiently long (which is typically the case when capturing photos or videos during the night), the insects appear as bright streaks on a dark background, due to motion-blur (Figures 1B, C). In annotation approaches 1 and 2 (and approach 3 for each individual video frame), the path of the motion-blur is annotated using a polyline for motion-blurs which are fully visible within the frame of the camera, or an enclosing circle when the motion-blur is partially occluded or out-of-frame. The geometries of the polyline annotations are later used by the wingbeat

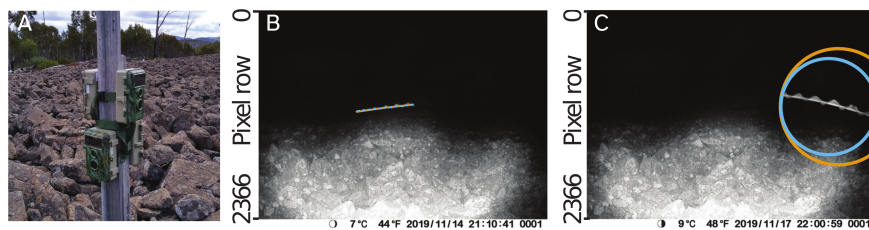


FIGURE 1

Example images showing data collection procedures used in this study. (A). Wildlife cameras (BlazeVideo, model SL112) were set to capture still photos and/or videos on timers and were deployed at the study sites known for their abundance of target flying insects. Typically, cameras were placed facing the sky, but could also be placed on an elevated mount, such as a post (pictured). (B). Motion blurs of moths captured by the cameras were marked with a polyline annotation. Manual annotation made in VIA (23) is shown in orange, and the annotation made by our automated procedure is shown in blue (although since both annotations are very similar, they overlap and only the blue annotation is visible). (C). Circular or point annotations were used for images of moths whose motion blurs were not fully contained within the frame of the camera, or where the length of the motion blur was too short to see the moth's wingbeat (latter case not shown). Manual annotation made in VIA (23) is shown in orange, and the annotation made by our automated procedure is shown in blue.

frequency measurement procedure (defined in [Supplementary Material S2](#)) and by the procedure for tracking multiple insects between video frames, described below.

2.2.1 Approach 1: manual annotation of still images

Images are manually annotated for flying moths using VIA (23). A step-by-step guide to performing the annotations is provided in Camfi's documentation (<https://camfi.readthedocs.io/>). Examples of polyline and circle annotations are displayed in [Figures 1B, C](#) (orange annotations).

We manually annotated 42420 images for flying insects. We reserved 250 images which contained annotations as a test set, and the rest were used for training the automatic annotation model.

2.2.2 Approach 2: automated annotation of still images

Although the process of manually annotating the images is simple to undertake, it is also time-consuming, particularly for large volumes of images. For large-scale studies, it may be desirable to use automated annotation, either by itself or in conjunction with manual annotation. To that end, we have developed an automatic annotation tool, which is included with Camfi, and used by running `camfi annotate` from the command-line. The automatic annotation relies on Mask R-CNN (14), a state-of-the-art deep learning framework for object instance segmentation. The tool operates on VIA project files, allowing it to serve as a drop-in replacement for manual annotation. The tool also allows the annotations it generates to be loaded into VIA and manually edited if required. Examples of the annotations made by the automated procedure are displayed in [Figures 1B, C](#) (blue annotations).

A full description of the automated annotation procedure, including training, inference, and validation, is provided in the [Supplementary Material S1](#). In summary, the procedure for model training uses the manual polyline annotations to create segmentation masks and bounding boxes which are used together

with the images from which those annotations come from as training data for the Mask R-CNN model. Automated annotation uses the outputs of Mask R-CNN model inference (which are again segmentation masks and bounding boxes) to infer the geometries of polyline or circle annotations.

It should be noted that the automated annotation method uses polynomial regression to infer polyline annotations from the segmentation masks produced by Mask R-CNN (the order of the polynomial can be configured at run time). This is predicated on the target insects' flight trajectories (within the course of a single exposure, e.g. 1/9 s) being able to be modelled with simple curves. In our experience, 2nd order polynomials are sufficient, however a higher order may be required for insects which fly very tortuous paths.

We have pre-trained the annotation model on the manual annotations we made using approach 1 (above). Largely speaking, these annotations are of Bogong moths (a noctuid with ca. 5 cm wingspan). We expect the automated annotation to therefore work well for moths which are of a similar size and have a similar appearance while in flight. To simplify training of the model to target other species, we have implemented a tool which automates the training process, and this is described in the [Supplementary Material S1](#). This tool is packaged with Camfi and is used by running `camfi train` from the command-line.

2.2.3 Approach 3: automated video annotation

An advantage of Camfi is its flexibility regarding the temporal resolution of data collection. Depending on the research question, cameras can be set to capture an image at relatively long intervals, on the order of minutes, or they can be set to capture images at a very high rate, which in the case of video clips is on the order of hundredths of a second (typically 25-60 frames per second). However, when analysing Camfi data which have been obtained from high-rate captures (namely, videos), individuals will be detected multiple times, since each moth will be seen in each of many consecutive video frames as they pass by the camera. This results in detection counts being inflated by insects which have lower angular velocities relative to others from the perspective of the

camera, and therefore spend more time in-frame. Therefore, to facilitate the use of videos by Camfi, we need to be able to track observations of individuals in a sequence of video frames, so we can count each individual only once.

In the following sections, we introduce an extension to Camfi which enables analysis of video data. This includes proper handling of video files, as well as tracking of individuals through consecutive frames. In addition to ensuring individual insects are only counted once per traversal of the camera's field of view, the new method allows for measurement of the direction of displacement of insects as they travel through the air.

2.3 Multiple object tracking

Multiple object tracking is a challenging problem which arises in many computer vision applications, and which has been approached in a variety of different ways (reviewed by 24).

A common approach to multiple object tracking is “detection-based tracking” (also known as “tracking-by-detection”), in which objects are detected in each frame independently, and then linked together using one of a number of possible algorithms. Typically, this requires the use of a model of the motion of the objects to be tracked, along with a method which uses the model of motion to optimise the assignment of detections to new or existing trajectories. In many approaches, the modelled motion of tracked objects is inferred by combining information about the position of the objects in multiple frames. An obvious challenge arises here because the model of motion requires reliable identity information of objects detected in multiple frames, whereas the identity of the objects usually must be inferred from their motion (including their position). This circular dependency—between the inference of object identity and the model of motion—can be dealt with in a number of ways, including probabilistic inference via a Kalman filter (25) or a particle filter (e.g. 26), or through deterministic optimisation using a variety of graph-based methods.

Our approach to multiple object detection removes the requirement of an explicit model of motion entirely, by utilising two peculiar properties of the Camfi object detector. The first of these properties is that the Camfi detector obtains information about the motion of the flying insects it detects from the motion blurs the insects generate, which it stores in the form of a polyline annotation. Since this information is obtained from a single image, and therefore a single detection, it does not depend on the identity of the insect, solving the previously mentioned circular dependency problem. The second property is that the Camfi detector is robust to varying exposure times, owing to the fact it has been trained on images with a variety of exposure times. This in turn means that the detector is robust to the length of the insects' motion blurs. Ultimately, these two properties, along with the fact that the insects appear as light objects on a dark background, mean that it is possible to use the Camfi detector to make a single detection of an individual insect traversing multiple consecutive frames. Thus, the trajectories can simply be formed using bipartite graph matching of

overlapping polyline annotations, using only information provided by the detections themselves, using the method described below.

2.4 Automated flying insect tracking

The algorithm for tracking flying insects in short video clips uses a detection-based tracking paradigm, relying heavily on the Camfi flying insect detector described above. An example of the sequence of steps taken by the tracking algorithm described in this section is illustrated in Figure 2. For brevity, the example shows the algorithm operating on three frames only, however the algorithm can operate on any number of frames, up to the memory constraints of the computer it is running on.

First, the video frames are prepared for flying insect detection. A batch of frames is loaded into memory (e.g. Figures 2A–C, although typically this would be a video clip). The maximum image of each sequential pair of frames is then calculated by taking the maximum (brightest) value for each pixel between the two frames (Figures 2D, E). This produces images with lengthened motion blurs of the in-frame flying insects, approximating the images which would be obtained if the exposure time of the camera were doubled. Importantly, the motion blurs of an individual insect in consecutive time-steps overlap each other in these maximum images.

Detection of flying insects is performed on the maximum images using the Camfi detector, producing candidate annotations of insect motion blurs to be included in trajectories (Figures 2F, G). The Camfi detector produces polyline annotations which follow the respective paths of the motion blurs of flying insects captured by the camera. Because the motion blurs of individual insects overlap in consecutive frames, so too do the annotations of those blurs (e.g. Figure 2H). This enables the construction of trajectories by linking overlapping sequential detections.

Detections in successive time-steps are linked by solving the linear sum assignment problem using the modified Jonker-Volgenant algorithm with no initialisation, as described by Crouse (27). In order to do this, a formal definition of the cost of linking detections is required. We call this cost the “matching distance”, which we denote by d_M . Consider two polyline annotations P_a and P_b , which are sequences of line segments defined by the sequences of vertices $(a_i)_{i=0}^{m-1}$ and $(b_j)_{j=0}^{n-1}$, respectively, where $a_i, b_j \in \mathbb{R}^2$. We define $d_M(P_a, P_b)$ as the second smallest element in $\{d(a_0, P_b), d(a_{n-1}, P_b), d(b_0, P_a), d(b_{m-1}, P_a)\}$, where $d(x, P)$ is the Euclidean distance from a point $x \in \mathbb{R}^2$ to the closest point in a polyline $P \subset \mathbb{R}^2$. This definition of d_M is efficient to compute, and allows us to discriminate between pairs of detections which come close to each other by chance (perhaps at very different angles) and pairs of detections which closely follow the same trajectory (i.e. roughly overlap each other).

After solving the assignment problem, a heuristic is applied to reduce spurious linking of detections into trajectories, where links with d_M values above a specified threshold are removed. Trajectories are built across the entire batch of frames by iteratively applying the detection linking procedure for each

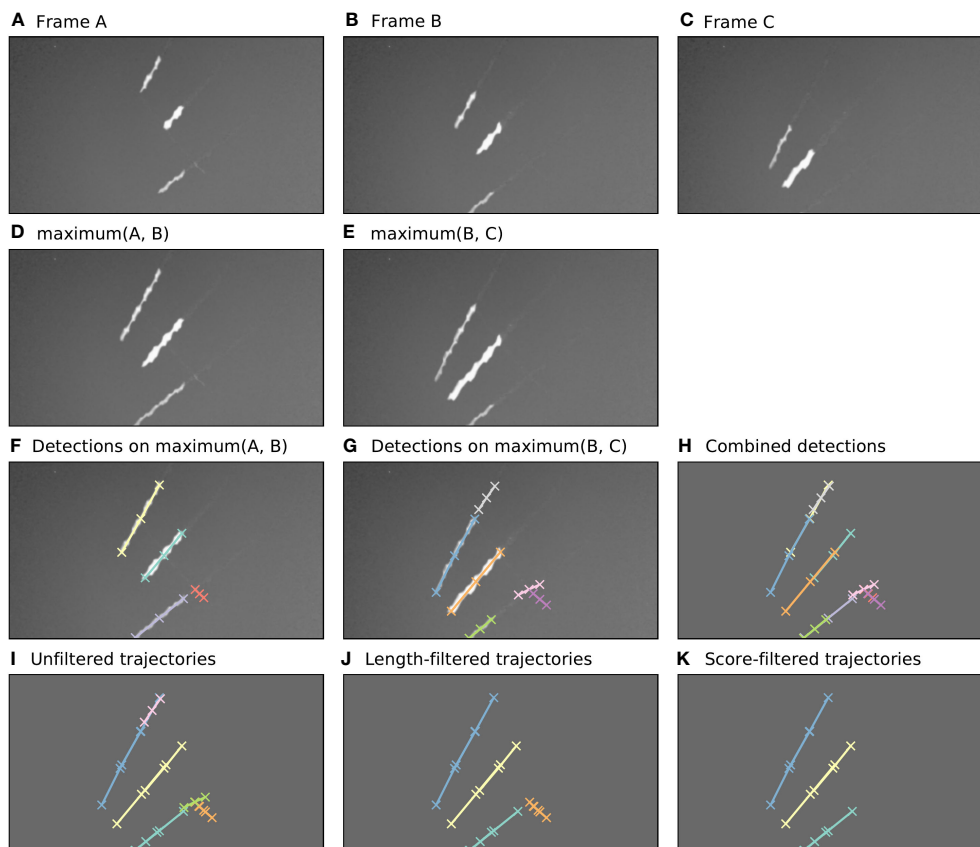


FIGURE 2

Automatic annotation is performed by Camfi on the maximum image of each pair of consecutive frames, allowing trajectories to be built from overlapping detections. Here, an example of this process is shown for three consecutive video frames. (A–C). Three consecutive video frames containing multiple flying insects. (D, E). The maximum image of each sequential pair of frames. (F, G). Flying insects are detected in the two-frame maximum images using Camfi. (H). Detections from (F, G) together on a plain background. (I). Detections from sequential time-steps are combined into trajectories using bipartite graph matching on the degree of overlap between the detections. (J). Trajectories containing fewer than three detections are removed. (K). Finally, trajectories are filtered by mean detection score (trajectories with mean detection score lower than 0.8 are removed).

consecutive pair of time-steps (Figure 2I). Trajectories containing fewer than three detections are removed (Figure 2J), as are trajectories with low mean detection scores (Figure 2K). The threshold for what is considered a low detection score can be set by the user (e.g. a value of 0.8 might be reasonable). When analyses relating to flight track directions are required, an additional filtering step can be applied to constrain analysis to detections inside a circular region of interest within the frame. This eliminates directional bias arising from the non-rotationally symmetrical rectangular shape of the video frames.

Diagnostic plots of tracking performance over an entire short video clip can be made by taking the maximum image of the entire video clip, and plotting the detected trajectories as a single image using a different colour for each trajectory (e.g. Figure 3). For example, we can see good performance of the tracking procedure in Figure 3A, where all trajectories except one appear to have been correctly built. The one exception is an insect close to the centre of that figure which appears to have had its trajectory split in three parts (seen as three different coloured segments), most likely due to occlusion by another insect. Figure 3B shows the result of constraining these trajectories to a circular region of interest to remove directional bias (in this case, this happened to solve the aforementioned split trajectory, but only by

coincidence—the orange and purple tracks were removed for overlapping the edge of the circle).

2.5 Implementation

Our implementation of Camfi and its associated tools is written in Python 3.9 (Python Software Foundation, <https://www.python.org/>). The latest version of Camfi relies on (in alphabetical order): bces 1.0.3 (28), exif 1.3.1 (29), imageio 2.9.0 (30), Jupyter 1.0.0 (31), Matplotlib 3.4.2 (32), NumPy 1.21.1 (33), Pandas 1.3.0 (34), Pillow 8.3.1 (35), pydantic 1.8.2 (36), Scikit-image 0.18.2 (37), Scikit-learn 0.24.2 (38), SciPy 1.7.0 (39), Shapely 1.7.1 (40), skyfield 1.39 (41), Statsmodels 0.12.2 (42), strictyaml 1.4.4, PyTorch 1.9.0 (43), TorchVision 0.10.0 (44), and tqdm 4.61.2 (45).

Camfi is open source and available under the MIT license. The full source code for the latest version of Camfi and all analyses presented in this paper are provided at <https://github.com/J-Wall/camfi>. The documentation for Camfi is provided at <https://camfi.readthedocs.io/>. Camfi is under active development and we expect new features and new trained models to be added as new versions of camfi are released from time to time. All analyses

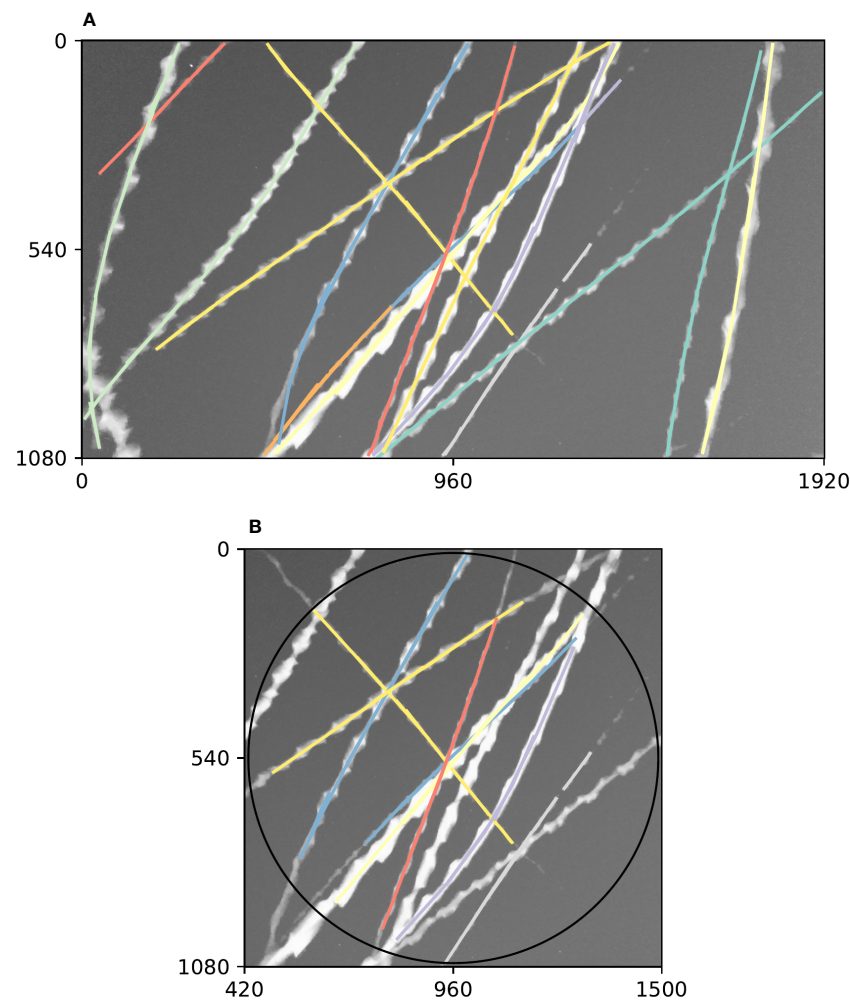


FIGURE 3

Example summary of trajectories followed by insects flying past a camera during a 5 s video clip. Axes on both plots show pixel row and column numbers. (A). Maximum (brightest) value of each pixel across every frame in the clip with annotations overlaid. Visible bright streaks are made by the motion blurs of Bogong moths flying past the camera. The colour of an annotation indicates its membership in a unique trajectory, as predicted by our method. (B). Annotations constrained to circular region of interest. Using only these trajectories eliminates directional bias resulting from the non-rotationally symmetrical rectangular shape of the frame. Black circle shows region of interest.

presented in this paper were done using Camfi 2.1.4, which is permanently available from the Zenodo repository <https://doi.org/10.5281/zenodo.5242596> (46).

3 Results

Automatic annotation performance was evaluated using a test set of 250 images, as well as the full set of 42420 images. Contour plots of evaluation metrics for both sets are presented in Figure 4. The metrics presented are prediction score vs. intersection over union, polyline Hausdorff distance, and polyline length difference (Figures 4B–D, respectively; see Supplementary S3 for definitions of these terms). These plots show similar performance on both the full image set (42420 images) and the test set (250 images), indicating that the model did not suffer from overfitting. Furthermore, they show that prediction scores for matched annotations (automatic annotations which were successfully matched to annotations in the

manual ground-truth dataset) tended to be quite high, as did the intersection over union of those annotations, while both polyline Hausdorff distance and polyline length difference clustered relatively close to zero. The precision-recall curves of the automatic annotator (Figure 4E) show similar performance between the image sets and show a drop in precision for recall values above 0.6. Training was completed in less than 2 h (Figure 4A) on a machine with two 8-core Intel Xeon E5-2660 CPUs running at 2.2GHz and a Nvidia T4 GPU, and inference took on average 1.15 s per image on a laptop with a 6-core Intel Xeon E-2276M CPU running at 2.8GHz and a Nvidia Quadro T2000 GPU.

4 Discussion

This paper demonstrates the utility of inexpensive wildlife cameras for the long-term population monitoring and observation of flying behaviour in flying insects. We do not

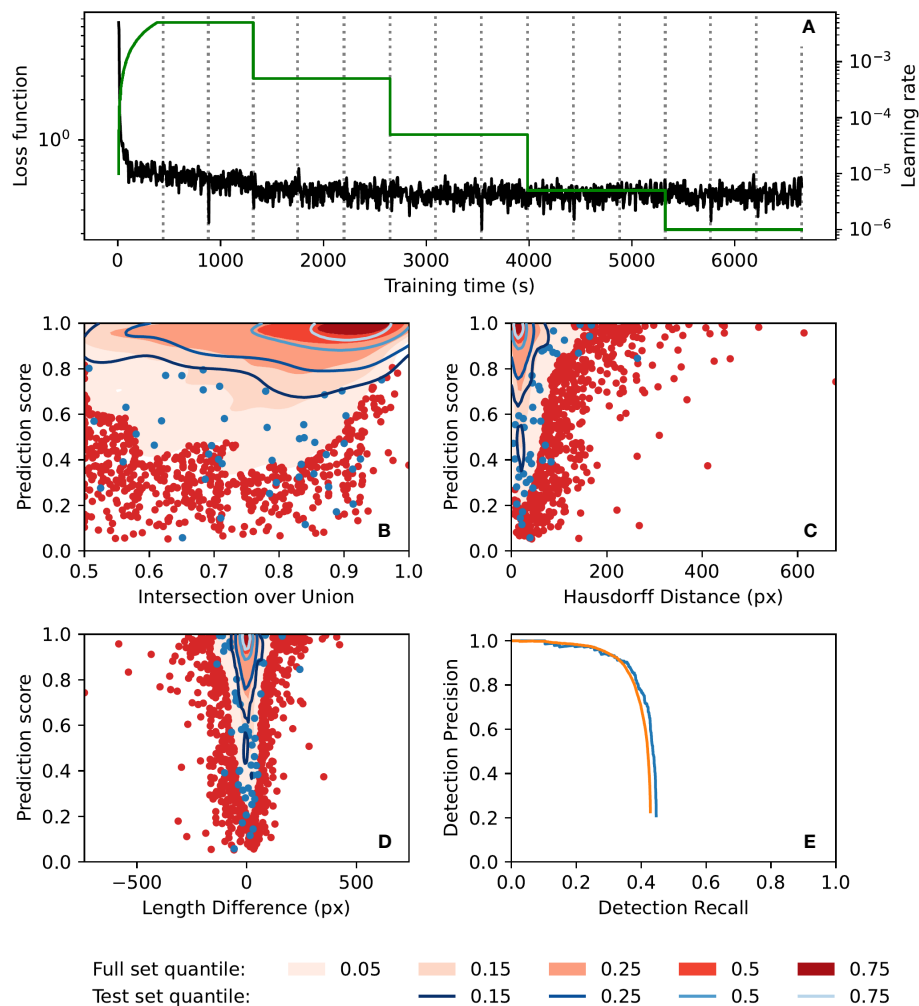


FIGURE 4

Automatic annotation evaluation plots. (A) Automatic annotation model training learning rate schedule (green) and loss function (black) over the course of training. Epochs (complete training data traversal) are shown with dotted vertical lines. (B–E). Similar performance was seen for both the full 42420-image set (red) and the test 250-image set (blue). Gaussian kernel density estimate contour plots of prediction score vs. (B) bounding box intersection over union, (C) polyline Hausdorff distance, and (D) polyline length difference, for both image sets. Contours are coloured according to density quantile (key at bottom of figure). In each plot, data which lie outside of the lowest density quantile contour are displayed as points. (E) Motion blur detection precision-recall curve, generated by varying prediction score threshold. The precision-recall curve for the set of images which had at least one manual annotation is shown in orange.

expect this method to completely replace other approaches for monitoring insects, such as trapping, which enables precise measurement of biodiversity and positive identification of species. Likewise, it will not completely replace other remote sensing approaches, such as radar and lidar, which facilitate detecting targets at long distances. However, it is clear that this method has significant potential for complementing these other approaches, and in certain circumstances, replacing them. For instance, in comparison to these other approaches, this method is particularly suited to monitoring assemblages of known species in remote areas, especially when it is known that the target insects are low-flying. An advantage of the presented method over trapping is that much greater temporal resolution is gained, and the sampling rate can easily be adjusted depending on the research question, simply by changing the settings on the cameras. This is in contrast to trapping studies, where typically only one measurement of abundance can be

recorded per visit to the trap by the researcher. This provides an opportunity to use the present method to answer a variety of ethological research questions which may not be approachable with previous methods.

This paper has presented a method for monitoring nocturnal flying insects, however there is no reason it could not be used for diurnal species as well, provided care is taken with regard to the placement of cameras. Namely, it would be important to have a relatively uniform background (such as the sky) in order to be able to see insects in the images during the day. In this case, the infra-red flash of the cameras would not be used and the insects would appear as dark objects on a light background. During the day, the exposure time of the cameras is much shorter than at night, so it would be impossible to use this method to measure wingbeat frequencies of day-flying insects. However, in some cases it may be possible to identify day-flying insects in the images directly. It may also be

possible to recreate the type of images seen during the night in any lighting conditions by retrofitting the cameras with long-pass infrared filters, neutral density filters, or a combination of both.

A key advantage of the present method over other approaches is that it can be readily scaled to large monitoring studies or programs, thanks to the low cost of implementation and the inclusion of the tool for automatic annotation of flying insect motion blurs. It is expected that studies implementing this method for target species which substantially differ in appearance from Bogong moths when in flight (and where the use of automatic annotation is desired) may have to re-train the Mask R-CNN instance segmentation model. We believe that the tools we have implemented make that process highly accessible.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: <https://doi.org/10.5281/zenodo.5194496>.

Ethics statement

Ethical review and approval was not required for the study on animals in accordance with the local legislation and institutional requirements.

Author contributions

JW, JZ, and EW conceived the ideas. JW, TR, and DD conducted the fieldwork. TR and JW performed the manual annotations. JW wrote the software. JW, BB, and DD performed the analyses. JW wrote the first draft of the manuscript with input from EW. and all authors edited the manuscript until completion. All authors contributed to the article and approved the submitted version.

Funding

EW and JW are grateful for funding from the European Research Council (Advanced Grant No. 741298 to EW), and the

Royal Physiographic Society of Lund (to JW). JW is thankful for the support of an Australian Government Research Training Program (RTP) Scholarship.

Acknowledgments

We thank Drs. Ryszard Maleszka, Linda Broome, Ken Green, Samuel Jansson, Alistair Drake, Benjamin Amenuvege, Mr. Benjamin Mathews-Hunter, and Ms. Dzifa Amenuvege Wallace for invaluable collaboration and assistance, and Drs. John Clarke and Francis Hui for useful discussions relating to the wingbeat frequency analyses.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/finsc.2023.1240400/full#supplementary-material>

References

- Drake VA, Farrow RA. A radar and aerial-trapping study of an early spring migration of moths (Lepidoptera) in inland New South Wales. *Aust J Ecol* (1985) 10:223–3. doi: 10.1111/j.1442-9993.1985.tb00885.x
- Knuff AK, WiNiger N, Klein A-M, Segelbacher G, Staab M. Optimizing sampling of flying insects using a modified window trap. *Meth Ecol Evol* (2019) 10:1820–5. doi: 10.1111/2041-210X.13258
- Beck J, Linsenmair KE. Feasibility of light-trapping in community research on moths: Attraction radius of light, completeness of samples, nightly flight times and seasonality of Southeast-Asian hawkmoths (Lepidoptera: Sphingidae). *J Res Lepidoptera* (2006) 39:18–37. doi: 10.5962/p.266537
- Infusino M, Brehm G, Di Marco C, Scalercio S. Assessing the efficiency of UV LEDs as light sources for sampling the diversity of macro-moths (Lepidoptera). *Euro. J Entomol* (2017) 114:25–33. doi: 10.14411/eje.2017.004
- Athanassiou CG, Kavallieratos NG, Mazomenos BE. Effect of trap type, trap color, trapping location, and pheromone dispenser on captures of male *Palpita unionalis* (Lepidoptera: Pyralidae). *J Econ Entomol* (2004) 97:321–9. doi: 10.1093/jee/97.2.321
- Laurent P, Frérot B. Monitoring of European corn borer with pheromone-baited traps: Review of trapping system basics and remaining problems. *J Econ Entomol* (2007) 100:1797–807. doi: 10.1093/jee/100.6.1797
- Eastwood E. *Radar ornithology*. Methuen, London, United Kingdom (1967).
- Gauthreaux JSA, Belser CG. Radar ornithology and biological conservation. *Auk* (2003) 120:266–77. doi: 10.1093/auk/120.2.266
- Riley JR. Remote sensing in entomology. *Ann Rev Entomol* (1989) 34:247–71. doi: 10.1146/annurev.en.34.010189.001335

10. Drake VA, Reynolds DR. *Radar entomology: Observing insect flight and migration*. Wallingford, Oxfordshire, United Kingdom: Cabi Press (2012).
11. Chapman JW, Drake VA, Reynolds DR. Recent insights from radar studies of insect flight. *Ann Rev Entomol* (2011) 56:337–56. doi: 10.1146/annurev-ento-120709-144820
12. Brydegaard M, Malmqvist E, Jansson S, Larsson J, Török S, Zhao G. The Scheimpflug lidar method. In: *Lidar remote sensing for environmental monitoring 2017*. San Diego, California, United States: International Society for Optics: Photonics (2017). p. 104060I. doi: 10.1117/12.2272939
13. Brydegaard M, Jansson S. Advances in entomological laser radar. *J Eng.* (2019) 2019:7542–5. doi: 10.1049/joe.2019.0598
14. He K, Gkioxari G, Dollár P, Girshick R. (2017). Mask R-CNN, in: *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy: IEEE. pp. 2961–9. doi: 10.1109/iccv.2017.322
15. Wallace JRA, Dreyer D, Reber T, Khaldy L, Mathews-Hunter B, Green K, et al. Camera-based monitoring of flying insects in the wild (Camfi). II. Flight behaviour and long-term population monitoring of migratory Bogong moths in Alpine Australia. *Front Insect Sci* (2023) 3. doi: 10.3389/finsc.2023.1230501
16. Green K. The transport of nutrients and energy into the Australian Snowy Mountains by migrating Bogong moths *Agrotis infusa*. *Austral Ecol* (2011) 36:25–34. doi: 10.1111/j.1442-9993.2010.02109.x
17. Warrant EJ, Frost B, Green K, Mouritsen H, Dreyer D, Adden A, et al. The Australian Bogong moth *Agrotis infusa*: A long-distance nocturnal navigator. *Front Behav Neurosci* (2016) 10:77. doi: 10.3389/fnbeh.2016.00077
18. Dreyer D, Frost B, Mouritsen H, Günther A, Green K, Whitehouse M, et al. The Earth's magnetic field and visual landmarks steer migratory flight behavior in the nocturnal Australian Bogong moth. *Curr Biol* (2018) 28:2160–6. doi: 10.1016/j.cub.2018.05.030
19. Adden A, Wibrand S, Pfeiffer K, Warrant E, Heinze S. The brain of a nocturnal migratory insect, the Australian Bogong moth. *J Comp Neur.* (2020) 528:1942–63. doi: 10.1002/cne.24866
20. Mansergh I, Heinze D. Bogong moths “*Agrotis infusa*,” soil fertility and food chains in the Australian alpine region, and observations concerning the recent population crash of this iconic species. *Vict. Natural.* (2019) 136:192. doi: 10.3316/informit.903350783919719
21. Green K, Caley P, Baker M, Dreyer D, Wallace J, Warrant EJ. Australian Bogong moths *Agrotis infusa* (Lepidoptera: Noctuidae), 1951–2020: Decline and crash. *Austral Entomol* (2021) 60:66–81. doi: 10.1111/aen.12517
22. Sánchez-Bayo F, Wyckhuys KAG. Worldwide decline of the entomofauna: A review of its drivers. *Biol Cons.* (2019) 232:8–27. doi: 10.1016/j.biocon.2019.01.020
23. Dutta A, Zisserman A. (2019). The VIA annotation software for images, audio and video, in: *MM '19: Proceedings of the 27th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery New York NY United States. doi: 10.1145/3343031.3350535
24. Luo W, Xing J, Milan A, Zhang X, Liu W, Kim TK. Multiple object tracking: A literature review. *Art. Intell* (2021) 293:103448. doi: 10.1016/j.artint.2020.103448
25. Reid D. An algorithm for tracking multiple targets. *IEEE Trans Auto. Cont.* (1979) 24:843–54. doi: 10.1109/TAC.1979.1102177
26. Breitenstein MD, Reichlin F, Leibe B, Koller-Meier E, Van Gool L. (2009). Robust tracking-by-detection using a detector confidence particle filter, in: *2009 IEEE 12th International Conference on Computer Vision*. Kyoto Japan: IEEE. pp. 1515–22.
27. Crouse DF. On implementing 2D rectangular assignment algorithms. *IEEE Trans Aerosp. Electron Sys.* (2016) 52:1679–96. doi: 10.1109/TAES.2016.140952
28. Nemmen RS, Georganopoulos M, Guiriec S, Meyer ET, Gehrels N, Sambruna RM. A universal scaling for the energetics of relativistic jets from black hole systems. *Science* (2012) 338:1445–8. doi: 10.1126/science.1227416
29. Thieding T, ArgiesDario, Saunders J, RKrahl, chbndrhns, Monzel R. (2021). exif. PyPi.
30. Silvester S, Tanbakuchi A, Müller P, Nunez-Iglesias J, Harfouche M, Klein A, et al. *imageio/imageio v2.9.0*. Geneva, Switzerland: Zenodo (2020). doi: 10.5281/zenodo.4972048
31. Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, et al. Jupyter notebooks - a publishing format for reproducible computational workflows. In: Loizides F, Schmidt B, editors. *Positioning and power in academic publishing: players, agents and agendas*. Amsterdam, Netherlands: IOS Press (2016). p. 87–90. doi: 10.3233/978-1-61499-649-1-87
32. Hunter JD. Matplotlib: A 2D graphics environment. *IEEE Ann Hist. Comp* (2007) 9:90–5. doi: 10.1109/MCSE.2007.55
33. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. *Nature* (2020) 585:357–62. doi: 10.1038/s41586-020-2649-2
34. McKinney W. (2010). Data structures for statistical computing in python, in: *Proc. 9th Python in Science Conf*, Austin, TX, United States. pp. 51–6. doi: 10.25080/Majora-92bf1922-00a
35. Van Kemenade H, wiredfool, Murray A, Clark A, Karpinsky A, Baranović O, et al. *python-pillow/pillow: 8.3.1*. Geneva, Switzerland: Zenodo (2021). doi: 10.5281/zenodo.5076624
36. Colvin S, et al. (2021). PyPi.
37. Van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, et al. Scikit-image: Image processing in python. *PeerJ* (2014) 2:e453. doi: 10.7717/peerj.453
38. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in python. *J Mach Learn Res* (2011) 12:2825–30.
39. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nat Meth* (2020) 17:261–72. doi: 10.1038/s41592-019-0686-2
40. Gillies S. *Shapely: Manipulation and analysis of geometric objects*. (2007). Available at: <https://github.com/shapely/shapely>.
41. Rhodes B. *Skyfield: High precision research-grade positions for planets and earth satellites generator*. (2019). Houghton Michigan, United States: Astrophysics Source Code Library record ascl:1907.024.
42. Seabold S, Perktold J. (2010). Statsmodels: Econometric and statistical modeling with python, in: *Proc. 9th Python in Science Conf*. Austin, TX, United States. p. 61. doi: 10.25080/Majora-92bf1922-011
43. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, dAlché-Buc F, Fox E, Garnett R, editors. *Advances in neural information processing systems*, vol. 32. Vancouver, BC, Canada: Curran Associates, Inc. (2019). p. 8024–35.
44. Marcel S, Rodriguez Y. (2010). Torchvision: The machine-vision package of torch, in: *MM '10: Proceedings of the 18th ACM international conference on Multimedia*, Firenze Italy: Association for Computing Machinery New York NY United States. pp. 1485–8. doi: 10.1145/1873951.1874254
45. da Costa-Luis C, Larroque SK, Altendorf K, Mary H, Sheridan R, Korobov M, et al. *tqdm: A fast, extensible progress bar for Python and CLI*. Geneva, Switzerland: Zenodo (2021). doi: 10.5281/zenodo.5109730
46. Wallace JRA. *J-wall/camfi: version 2.1.3*. Geneva, Switzerland: Zenodo (2021). doi: 10.5281/zenodo.5194496