Received: 24 June 2025 | Accepted: 18 March 2026

DOI: 10.1111/2041-210x.70304

BRITISH ECOLOGICAL SOCIETY | Methods in Ecology and Evolution

PRACTICAL TOOLS

AutoPollS: A tool for automated monitoring of pollinators using deep learning

Matthew A.-Y. Smith[1,2] | Olivia M. Bernauer[1,3] | Grace Melone[1] | Brett Graham[4,5] | Julia Wiessing[1] | Rafael Salas[1] | Jeremy Hemberger[1,6] | Abhishek Bhattacharyya[7] | Acacia T. S. Tang[1] | Elena Rojas[1] | Joshua San Miguel[7] | Brian J. Spiesman[8] | Claudio Gratton[1] | James D. Crall[1]

[1]Department of Entomology, University of Wisconsin-Madison, Madison, Wisconsin, USA; [2]Department of Biology, Illinois Institute of Technology, Chicago, Illinois, USA; [3]Department of Biology, University of Wisconsin-Eau Claire, Eau Claire, Wisconsin, USA; [4]Harvard University- Center for Brain Science, Cambridge, Massachusetts, USA; [5]Draft Product Development, Sudbury, Massachusetts, USA; [6]Department of Entomology, University of Minnesota, St Paul, Minnesota, USA; [7]University of Wisconsin-Madison, Department of Electrical and Computer Engineering, Madison, Wisconsin, USA and [8]Department of Entomology, Kansas State University, Manhattan, Kansas, USA

Correspondence
Matthew A.-Y. Smith
Email: msmith84@iit.edu

James D. Crall
Email: james.crall@wisc.edu

Funding information
Winslow Foundation; National Institute of Food and Agriculture, Grant/Award Number: 2022-67013-36275; Wisconsin Alumni Research Foundation

Handling Editor: Willem Bonnaffe

Abstract

1. Deep learning and computer vision hold enormous potential for automated monitoring of biodiversity, including pollinators and other insects. Efficient, scalable monitoring of insect pollinators is crucial given pollinators' role in supporting biodiversity and agricultural productivity amidst declining pollinator populations.
2. However, several practical challenges limit the broad adoption of automated monitoring techniques. Existing approaches often depend on passive image capture (e.g. timelapse) that can generate impractically large datasets (especially for large-scale monitoring) and rely on application-specific models that may not generalize well to novel contexts. This creates barriers to broader use in ecological applications where limited training data exist and underscores the need for flexible, reliable automated monitoring systems.
3. Here we introduce AutoPollS (Autonomous Pollinator Sampler), an open-source system for automated, image-based monitoring of flower-visiting insects. AutoPollS uses multi-camera monitoring using lightweight deep learning models in the field (to reduce data storage and processing) followed by higher performance detection and classification models offline. AutoPollS uses a modular camera system that can be adapted to different plant–pollinator communities.
4. We validate AutoPollS in two agroecosystems (sunflowers and apple orchards) and a multi-species alpine meadow. In all three systems, AutoPollS provided robust in-field monitoring using broadly trained models fine-tuned with only limited application-specific data, including high detection and species classification accuracy (>95%) in honey bees (*Apis* spp.) and bumble bees (*Bombus* spp.). However,

models did show reduced performance in real-world applications compared to training datasets—likely associated with image quality and curation—highlighting the challenges of monitoring in real-world ecological conditions.

5. Overall, our results demonstrate the utility of our approach in diverse ecological applications, including non-lethal monitoring of endangered species and characterizing pollinator responses to dynamic environmental conditions. Finally, we highlight the potential of this approach to bridge practical applications and model development by facilitating broader adoption. This could support generation of larger and more diverse datasets to improve model performance.

KEYWORDS

biodiversity monitoring, community ecology, conservation technology, edge computing, plant-pollinator interactions

1 | INTRODUCTION

Insect pollinators are essential to supporting biodiversity, global food production, and human health (Klein et al., 2007; Ollerton, 2017; Reilly et al., 2020); monitoring pollinator populations and their responses to novel anthropogenic environmental stressors is thus of global interest (Klein et al., 2007; Ollerton, 2017; Reilly et al., 2020; Yang et al., 2021). While automated monitoring with computer vision and deep learning is emerging as a powerful approach, large-scale automated pollinator monitoring poses several key challenges. For example, insect pollinators are highly diverse (Ollerton, 2017), small-bodied, and their activity and behaviour can vary dramatically in time and space (Olesen et al., 2008) or in response to dynamic environmental conditions (Corbet, 1990). This highlights the need for—and growing interest in (Woodard et al., 2020)—large-scale monitoring efforts to better characterize the dynamics of pollinator behaviour and ecology across space and time.

Scaling up traditional monitoring techniques to address this need, however, creates significant challenges. Existing techniques for pollinator monitoring rely heavily on lethal sampling, posing practical and ethical obstacles to large-scale monitoring (Gezon et al., 2015; Woodard et al., 2020). For example, the presence of endangered insect species, such as the Rusty Patched Bumblebee (*Bombus affinis* Cresson), can lead to restrictions on traditional sampling (Tepedino & Portman, 2021). While non-lethal approaches such as visual monitoring can substitute, these are generally labour-intensive, requiring significant taxonomic expertise and effort to identify pollinators during flight, thus limiting the scope of their application (Klaus et al., 2024).

In light of these challenges, machine learning and computer vision hold enormous potential for automated, non-lethal monitoring of pollinators and other insects (Ärje et al., 2020; Bjerge et al., 2022, 2023; Droissart et al., 2021; Geissmann et al., 2022; Høye et al., 2021; van Klink et al., 2022) with increasing accuracy and with increasing taxonomic breadth (Droissart et al., 2021; Geissmann et al., 2022; Høye et al., 2021; Tuia et al., 2022). However, many existing systems rely on continuous image capture systems (e.g. timelapse) which, while effective, can generate large datasets that pose significant challenges especially for large-scale or long-term monitoring (Ștefan et al., 2024), including bottlenecks and costs associated with storage, transfer, and analysis of image datasets. This is an acute issue for detecting flower-visiting insects, where pollinators may be visible for short durations (a few seconds) and visits occur sparsely in time, requiring substantial image storage.

Performing deep learning inference directly on devices deployed in the field for monitoring pollinators (i.e. 'edge' processing, Bjerge et al., 2025; Sittinger et al., 2024) is a promising emerging approach, allowing image processing in the field in 'real-time' and reducing data storage and offline processing requirements. While edge-compatible models are generally lightweight, fast and use low computational resources, they often come with a tradeoff of lower accuracy (Huang et al., 2022). This limitation may be compounded for applications where edge models are deployed in novel contexts (e.g. ecological communities where models are 'naive'). Existing pollinator monitoring approaches using edge processing also typically use custom models trained on static or standardized backgrounds (Høye et al., 2025) that—while facilitating adoption in different environments—may limit their application in direct monitoring of pollinator visitation to real flowers.

Effective and scalable monitoring of pollinators (and plant-pollinator interactions) will require deep learning models and hardware implementations that generalize well across taxonomic diversity and ecological contexts while effectively balancing the challenges of data storage with the accuracy of pollinator detection. Combining multiple, sequential models at different stages of the monitoring pipeline may help balance these tradeoffs in real-world applications. For example, edge detection (Bjerge et al., 2025) can be used to filter for insect-containing images, which can then be stored and passed to higher-resolution models offline, balancing efficient image collection in the field (via edge filtering) with higher accuracy (and potentially improved taxonomic resolution) classification offline.

Here we introduce AutoPollS (Autonomous Pollinator Sampler), a modular, open-source system that implements edge data filtering coupled with offline analysis to provide pollinator monitoring across diverse ecological contexts, without requiring modification of hardware or deep learning models between different ecological applications (Figure 1a). AutoPollS aims to balance data storage and accuracy challenges by integrating permissive edge-processing using single-class insect object detection in the field, followed by subsequent offline detection and taxonomic classification using higher-resolution models. Using field trials in three distinct ecological systems, we demonstrate that a combination of models trained on pre-existing data with limited fine-tuning can provide robust monitoring of insect visitation to flowering plants with diverse growth forms (from trees to low-stature forbs). We introduce new detection and classification models that build on previous work (Spiesman et al., 2021) to incorporate a broader taxonomic range of flower-visiting insects. Finally, we provide an integrated pipeline for image collection and processing to facilitate broad accessibility and adoption of these tools for pollinator ecologists.

2 | MATERIALS AND METHODS

2.1 | System overview

Built around the Raspberry Pi microcomputer, AutoPollS provides automated detection of flower-visiting insects across multiple cameras simultaneously (Figure 1b), using on-device (i.e. 'edge')

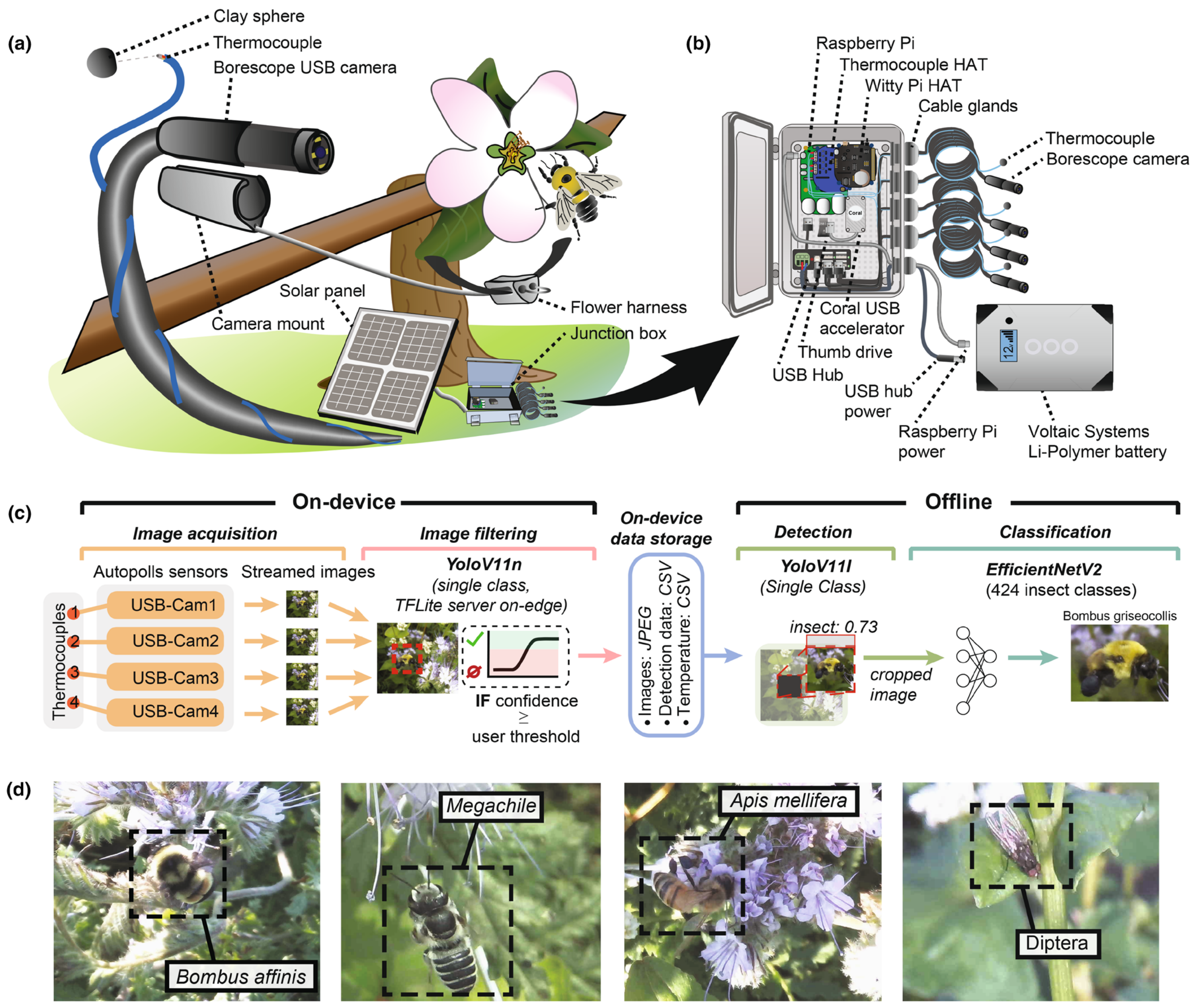

FIGURE 1 The AutoPollS system. (a) a schematic of the AutoPollS unit deployed to monitor apple blossoms, showing cameras and operative body temperature models (clay sphere and thermocouple). (b) AutoPollS unit components and integration (see Table S1 for details) (c) AutoPollS workflow from monitoring in the field to offline detection and classification. The AutoPollS system consists of four main steps: (1) image acquisition, (2) on-device filtering and then offline (3) object detection (YOLOv11), and (4) image classification (EfficientNetV2). (d) Example images captured by AutoPollS, with bounding boxes and inferred taxonomic classification labels displayed.

object-detection models implemented on commercially available components (Table S1). Local (on-device) storage of images when an insect is detected reduces data storage requirements in the field while allowing for subsequent analysis offline with higher-resolution detection and classification models.

2.2 | AutoPollS workflow

Detection and classification of insect pollinators occurs in four stages: (1) image acquisition and (2) on-device filtering using a lightweight edge detection model followed by (3) offline object detection and (4) image classification of stored images using higher accuracy, deep learning models (Figure 1c). Here, we trained three deep learning based models using a curated image dataset of bees and other flower-visiting insects, building on a previous image classification model for bumble bees (Spiesman et al., 2021): (1) an edge optimized single-class object-detection model (YOLOv11n) for edge filtering, (2) a higher-resolution offline single-class object detection model (YOLOv11l) for additional filtering, bounding box refinement, and image cropping, then finally (3) a classification model, EfficientNetV2 (Redmon et al., 2015; Tan & Le, 2021) for taxonomic identification—run on cropped images—that builds on a previous bumble bee image classification network (Spiesman et al., 2021). See Section 2.6 below for model details.

2.3 | Image acquisition: AutoPollS hardware

AutoPollS runs on a Raspberry Pi microcomputer (Pi 4B, quad-core ARM-Cortex-A72 and optimal performance with 2GB RAM or greater). The Raspberry Pi controls the AutoPollS firmware including image capture, edge model inference (see Section 2.4), temperature recording (optional), and local image storage to an external storage device using a combination of Python scripts controlled via Unix commands and Systemd services (Figure 1b; Figure S1). AutoPollS supports integration with two Raspberry Pi HATs: (1) the WittyPi (Table S1), adding a real-time clock & device power management and (2) the MCC-134 thermocouple HAT (Table S1), allowing the attachment of 4× thermocouple probes for local temperature recording (Figure 1b). AutoPollS can be powered directly from mains electricity, or via a battery and (optionally) solar panel for remote deployment. In the field trials described here, we used Voltaic Systems Li-Polymer batteries with solar panels.

In the default configuration, four 5MP Teslong™ USB waterproof borescope cameras connect to the Raspberry Pi via a 4-port USB hub (Table S1). The camera's 4.5 m semi-rigid and water-resistant cable allows for flexibility in placement and durability across a wide range of field conditions and plant growth forms (e.g., from low-growing herbs and forbs to large shrubs and trees). The system performance has been evaluated with the number of cameras ranging from 1 to 4 (Table S2).

2.4 | On-device image filtering: Object detection on the edge

Upon startup, AutoPollS automatically initializes software using configurable parameters (e.g., sample rate, confidence thresholds) and loads the on-device single-class detection model. AutoPollS will identify connected USB cameras automatically and provide a unique camera ID, then open a dedicated camera stream. The AutoPollS software creates a unique directory path for saving images based on the camera ID and current date. For each camera stream in sequence, an image is collected and added to a queue to be passed to the edge model. If a camera has an image in queue, it will not load new images from acquisition into memory. Images are taken from the queue, a copy is created, compressed to 300×300 pixels, and passed to the edge object-detection model. Preprocessing and inference for an image takes about ~1.1 s while running the edge model (see Section 2.6 below) on the Raspberry Pi CPU (Figure S2; Table S2, ~6 s interval for observation and inference per camera when four cameras are mounted). Outputs from the edge inference include: (1) bounding boxes for detected insects and (2) associated detection confidence scores for each bounding box. Images containing at least one bounding box with a confidence score exceeding a user-defined threshold (0–1) are then stored and saved to a local storage device in the original 5MP resolution. At a confidence threshold of 0, all images are stored and the system functions as a time-lapse camera The AutoPollS system writes stored images and associated processed data to a thumb drive mounted via USB. An edge-TPU processor (Coral USB Accelerator) can optionally be added to accelerate inference speed (Tables S1 and S2).

2.5 | Offline analysis (detection and classification)

All saved images collected from AutoPollS can then be passed to the offline, higher-resolution object-detection model with an image input size of 800×800 pixels for automated cropping and subsequently to the image classifier (see Section 2.6 below for model details). Specifically, all detected insects exceeding a user-defined threshold in the detection model are automatically cropped into separate images (using estimated bounding box coordinates) and passed to the classification model.

Crops from the full-resolution images are then resized to be 300×300 pixels and passed to the offline classification model (see Section 2.6 below), which assigns a confidence score of the cropped image belonging to each of the 424 classes in the model. The results from the classification were then merged with the results from the offline detector to give bounding box coordinates and detection confidence scores, along with the top classification result and classification confidence scores for each detected insect within an image. The offline detection and classification models were analysed using a custom-built workstation featuring an AMD Ryzen 75700G central processing unit (CPU), 64-GB RAM, Samsung 860 Evo Series 2-TB SSD, and a GeForce RTX 3080 Ti 12-GB GPU. Analysis scripts for offline detection and classification were executed in a Google Colab cloud-based environment using GPU resources (NVIDIA Tesla V100 GPU).

2.6 | Pretrained models

Here, we introduce three trained models used in the AutoPollS workflow: (1) an on-device YOLOv11n edge single-class detection model, (2) a larger offline YOLOv11l single-class detection model, and (3) an offline EfficientNetV2S taxonomic classification model. Each was trained primarily on broad, curated data sets of images containing flower-visiting insects combined with a small number of images generated from AutoPollS devices.

2.6.1 | Image data for model training

The detection models were single-class, while the classification model has 424 taxonomic classes (with a mix of taxonomic ranks from species to order; Table S3). The training, validation, and test image datasets comprised flower-visiting taxa occurring in the United States and were created by pooling curated images across open datasets from the Global Biodiversity Information Facility (GBIF) a substantial proportion of which comes from iNaturalist, images generated in the Spiesman lab at Kansas State University, and data contributed by the community science program the Wisconsin Bumble Bee Brigade. We supplemented the open datasets with a subset of annotated, AutoPollS-generated images to include representation of the domain of interest here. A total of 924 images (including both insect-containing and empty images) were added to the object-detection datasets, split into 805, 46, and 73 across the training, validation and test sets (Figure S3). A total of 4590 images were added to the classifier datasets—these images came from field studies not included in the analysis pipeline, featuring cranberry, apple, and wildflower fields (Figure S3). Images used in training, validation, or testing sets for the models were not used in the threshold optimization tuning or downstream analysis. The image dataset included classes of varying taxonomic resolution, including many common and well-represented bees to species (e.g. *Bombus* spp., and *Apis mellifera* L., other common insects) or genus level, as well as broader classes for some other flower-visiting insects (i.e., Syrphidae, other Diptera, Coleoptera, Lepidoptera, Hemiptera, Araneae, Formicidae and wasps). See Table S3 for details on included classes. Bounding boxes used for training and validation were generated using annotation software within the Roboflow online platform (Dwyer et al., 2022).

2.6.2 | Model performance metrics

Performance of the object detection and image classification models on test datasets collected through AutoPollS deployment was evaluated using precision (Equation 1) and recall (Equation 2).

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}, \tag{1}$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}. \tag{2}$$

These metrics are calculated by quantifying the number of True Positives, False Positives, and False Negatives in the results from model inference. The F1-score (Equation 3) is the harmonic mean of precision (Equation 1) and recall (Equation 2) and was chosen as the parameter to optimize against to maximize the number of true positives while minimizing the number of false positives in the results (Lipton et al., 2014).

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3}$$

For assessing the performance of the classifier model across field sites, we used an accuracy metric based on positive detections from the offline single-class detector (Equation 4).

$$\text{Accuracy} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative} + \text{False Positive}}. \tag{4}$$

Object-detection models

The object-detection models provide localization of the object in the image (as a bounding box). The YOLOv11 object-detection models output a localization confidence score that was used as the metric for confidence thresholding and data inclusion. We defined a True Positive to be an insect manually annotated as present and subsequently correctly identified through model inference with a confidence above the set threshold. We defined a False Positive as an inferred detection with a confidence above the set threshold, with no insect being annotated as present in the ground truth. We defined a False Negative as an insect present in the ground truth but not identified by the model inference results with a confidence above the set threshold.

Classification model

The image classifier outputs a vector of confidence scores containing the likelihood that the image belongs to each of the 424 classes. We defined a True Positive as an image being correctly classified into the same model class as the ground truth with a confidence above the set threshold. We defined a False Positive as an image being classified incorrectly with confidence above the set threshold. We defined a False Negative as when an insect was present in the ground truth dataset but was classified with a confidence below the set threshold by the model. The classification with the highest inference confidence is used to assess thresholding and classification accuracy. To further refine model performance, we evaluated the performance of the YOLOv11 models trained across two datasets: (1) supplemented with AutoPollS images, representing the deployment domain and (2) a dataset without images collected from AutoPollS (Figure S3).

2.6.3 | Model training

The object-detection models were initialized from a pretrained backbone on the COCO8 dataset using the Ultralytics (Jocher et al., 2023) and Pytorch (Paszke et al., 2019) python libraries. For YOLOv11n model training, 45,955 images were split into 80% training, 10% validation, and 10% test sets. Hyperparameters were set as: momentum: 0.937, weight decay: 0.0005, optimizer: Stochastic Gradient Descent with learning rate of 0.01. Training achieved a single-class (flower-visitor) precision of 0.955, recall of 0.905, F1 of 0.929, and mean average precision (mAP50) of 0.956 on the test dataset. The same protocol was used to train the offline YOLOv11l model. The YOLOv11l model training achieved a precision of 0.973 and a recall of 0.940, F1 of 0.956, and mAP50 of 0.972 on the test dataset.

The offline image classifier utilizes the EfficientNetV2S architecture and is trained on 424 classes of flower-visiting insects curated from the datasets listed above. The training dataset was prepared by using the YOLOv11l model to automatically crop images around the insects of interest. Each cropped image was then manually examined to ensure accurate crops of target taxa, resulting in a dataset of over 1.58 million images. The first approximately 10,000 bee images were annotated by hand and then used to train a model-assisted labelling feature within Roboflow. This provided annotations, but most had to be finetuned by hand. Once around 20,000 annotated bee images were confirmed, a new assisted labelling model was created, with manual adjustment required for ~25% of those annotations. Every image was checked by hand for correct placement of bounding boxes and adjusted if necessary. All non-bee images (spiders, ants, beetles, flies, true bugs, lepidoptera) were annotated by hand. Training data included 47,916 annotations across 44,096 images (~1.1 annotations per image). Images were then resized to 300 × 300 pixels and split into 80% training, 10% validation, and 10% test sets. The classification model was initiated on ImageNet weights and trained for 75 epochs to achieve a mean average precision of 0.95, recall of 0.95, and a F1 score of 0.95, and used the classification with the highest confidence for performance calculations. Taxon-specific precision, recall, and F1 scores are shown in Table S3.

During the training process, we weighted classes to account for class imbalance of taxa with lower representation and used image augmentation to increase generality to image variation and prevent overfitting. Images were standardized by rescaling pixel values to the range [0, 1]. Image augmentation techniques included random horizontal flipping, random rotation up to 100 degrees, random changes in height, width, shear, zoom, channel shift (range 0.2) and brightness shifts (range 0.9–1.1). The classification model was trained using the stochastic gradient descent (SGD) optimizer with an initial learning rate of 0.01 and categorical cross-entropy as the loss function. The learning rate was reduced by half when the validation loss plateaued during training (patience = 5).

2.7 | Field validation experiments

To test AutoPollS' performance in detecting flower-visiting insects in real-world applications, we deployed AutoPollS in three different systems. During these studies, all images were stored on the device while inference was running to provide an unbiased evaluation data set. Sampling permits were not required for any of the three sites.

2.7.1 | Sunflowers

AutoPollS units were deployed at the Arlington Agricultural Research Station in Dane County, Wisconsin in August 2022 in a field planted with one flowering focal crop (dwarf sunflowers, *Helianthus annuus* L.) alongside several wildflowers (*Chamaecrista fasciculata* (Michx.) Greene, *Coreopsis lanceolata* L., *Phacelia tanacetifolia* Benth., *Rudbeckia hirta* L., *Gaillardia pulchella* Foug., *Trifolium incarnatum* L. *and Fagopyrum esculentum* Moench), corn (*Zea mays* L.), and potatoes (*Solanum tuberosum* L.). Cameras were secured by attachment to a wooden stake and aligned with the same camera harnesses as above. We chose a single representative unit ($n = 4$ cameras) focused on sunflowers for 1 week in this test analysis, allowing a tractable amount of focal image annotation and a data set that was not overlapping with training and validation data. The AutoPollS ran continuously from 05:00 to 22:00, which yielded 32,206 images.

2.7.2 | Alpine wildflower meadow

Two AutoPollS units (4 cameras each, 8 cameras total) were deployed at the Colorado State University Mountain Campus located in Bellevue, Colorado for 8 days (July 7–July 15, 2023) across four wildflower species (*Penstemon* spp., *Oxytropis* spp., *Umbrellaria* spp., *and Potentilla* spp.). Each unit was programmed to monitor visitation at flowers from 05:00–21:00, yielding 166,195 images. Cameras were secured by attachment to a wooden stake and aligned with the same camera harnesses as above.

2.7.3 | Apple orchards

Five AutoPollS (four cameras each, 20 cameras total) were deployed in a commercial apple (*Malus domestica* Borkh.) orchard in Dane County, Wisconsin (USA) for 9 days (May 11–May 19, 2022) representing the total bloom period (Figures S4–S7). Each unit was programmed to monitor visitation at flowers for 15 min every hour from 08:00–20:00, yielding 273,999 images (Figure S4). Cameras were secured by wrapping semi-rigid camera cables around tree limbs and aligned over open flowers using a 3D-printed camera harness. Temperature was recorded via thermocouples connected to the thermocouple HAT (Figure 1a; Figure S6A) and was strongly correlated to air temperature at a local weather station (Figure S7B, Pearson-$r = 0.95$, $p = < 0.001$). Thermocouples were embedded in simple 'operative temperature' models (black polymer clay sphere, 2.5 g wet mass) that integrate the effects of air temperature and solar radiation on insect body temperature.

2.8 | Manual annotation of Autopolls collected data for ground truthing

Across the three field studies, all collected images were manually annotated offline for insect visitors. Initial annotations were generated by a single investigator (MS) and reviewed by at least one other investigator. For detection, rectangle bounding boxes were manually drawn around each detected flower-visiting arthropod that was above a size threshold of 120 × 120 pixel area and not substantially occluded. Annotations were recorded using open-sourced annotation software, LabelMe (Russell et al., 2008). These manual detections were labelled with classifications corresponding to insect taxa in the classification model used here (i.e. bees, wasps, beetles, and lepidopterans, but excluding ants, gnats, spiders, and other groups not included in the model output classes, Table S3). For some groups, manual classifications were provided at a higher taxonomic level (e.g. genus) than the classification model. For example, all *Andrena* species were aggregated to Andrenidae here (see Table S4 for corresponding classes used between the classification model and manual annotations). This yielded datasets of 842, 1614, and 2577 manually annotated visits for apple, sunflower and the alpine meadow, respectively. Data available via Zenodo and AWS: https://doi.org/10.5281/zenodo.19360800 (Smith et al., 2026).

2.9 | Confidence threshold calculation

To simulate various threshold combinations, a value was selected for each inference stage, and all images with confidence scores exceeding that threshold were passed to the next filtering step (offline detection or classification). The F1-score was calculated for the images that remained after the filtering by a parameter sweep (from 0 to 1 in 0.02 increments) for both filtering steps (see Figure S8 for F1 distributions across all datasets). To evaluate the generalizability of the chosen threshold across datasets, we selected one dataset (sunflower) to perform a 5-fold cross validation, with an 80:20 training, validation split for threshold calculations. The images included in the training split were randomly sampled and bootstrapped for 2500 iterations to estimate a distribution of the F1-scores and thresholds (Figure S9), yielding optimal (F1 maximum) threshold values of: edge model 0.46, offline detection model 0.73, classification model 0.11 (Figure S9B). These values were then used for all subsequent model evaluation, including the alpine meadow and apple datasets.

3 | RESULTS

We evaluated AutoPollS performance in the three field systems: (1) sunflower (Figure 2a–c), (2) alpine wildflower meadow (Figure 2d–f), and (3) apple (Figure 2g–i). We observed significant enrichment in F1 score across all sites through the filtering pipeline, yielding at minimum 15-fold improvement and final F1 values of 0.67, 0.62, and 0.62 for sunflower, alpine wildflower meadow, and apple orchard respectively. Overall, detection performance was robust across datasets; while insects were sparse in all datasets (0.33%–5.0% of images containing insect taxa of interest), on-device filtering enriched true positives in each dataset by approximately 10-fold (Figure 2a,d,g) while reducing data storage requirements by approximately 95% and retaining the majority of true positives: recalls of 0.83, 0.77 and 0.78 from on-device filtering for sunflower, alpine wildflower meadows, and apple orchard, respectively (Figure 2b,e,h). Precision was significantly improved by offline detection, with detection precisions of 0.61, 0.80 and 0.82 after offline processing for sunflower, alpine wildflower meadows, and apple orchard datasets, respectively. Overall classification accuracy was 88% in sunflowers, 89% for the alpine wildflower meadow, and 76% for apple (Figure 2c,f,i).

Model performance varied across datasets and taxonomic group: detection and classification for *Apis mellifera* and *Bombus* were highly accurate across datasets (>95% classification accuracy to the species level, Figure 2), while smaller and less common bees, as well as other taxonomic groups, showed higher rates of false negatives (i.e. not detected) or mis-classification (Figure 2c,f,i). Accuracy differences between taxa may result from many solitary bees being more difficult to detect given their small body size, as well as having lower representation in training sets used for the classification model.

We further validated the use of AutoPollS by comparing temperature-dependent activity patterns across taxa estimated from manually annotated versus model-inferred image data in the apple dataset (using the universal threshold values; Figure 2; Figure S9). We used data collected during the apple orchard study to quantify visitation patterns of different taxa across 7 days (Figure 3a,b). During this period, the community composition of flower-visiting insects shifted (i.e. from an abundance of Halictids and Andrenids to Diptera), corresponding to phenological progression of apple bloom (beginning with king bloom; Figures S5 and S6) and consistent with human observation (pers obs.). We paired the temperature recorded from the AutoPollS thermocouples with each visitation event to estimate a mean temperature when pollinators were detected (i.e. 'mean observation temperature', Figure 3b) for each taxonomic group (Figure 3b), which showed strong correlation between manually annotated (ground truth) data and model inference (Figure 3c, Pearson-$r = 0.95$, $p = 1.07 \times 10^{-3}$).

4 | DISCUSSION

Our results demonstrate the potential of AutoPollS—and more broadly the integration of edge and offline processing—as a generalizable and flexible framework for automated pollinator monitoring across ecological contexts. AutoPollS can provide monitoring of flower-visiting insects using modular hardware and broadly trained deep learning models in diverse ecological communities. AutoPollS

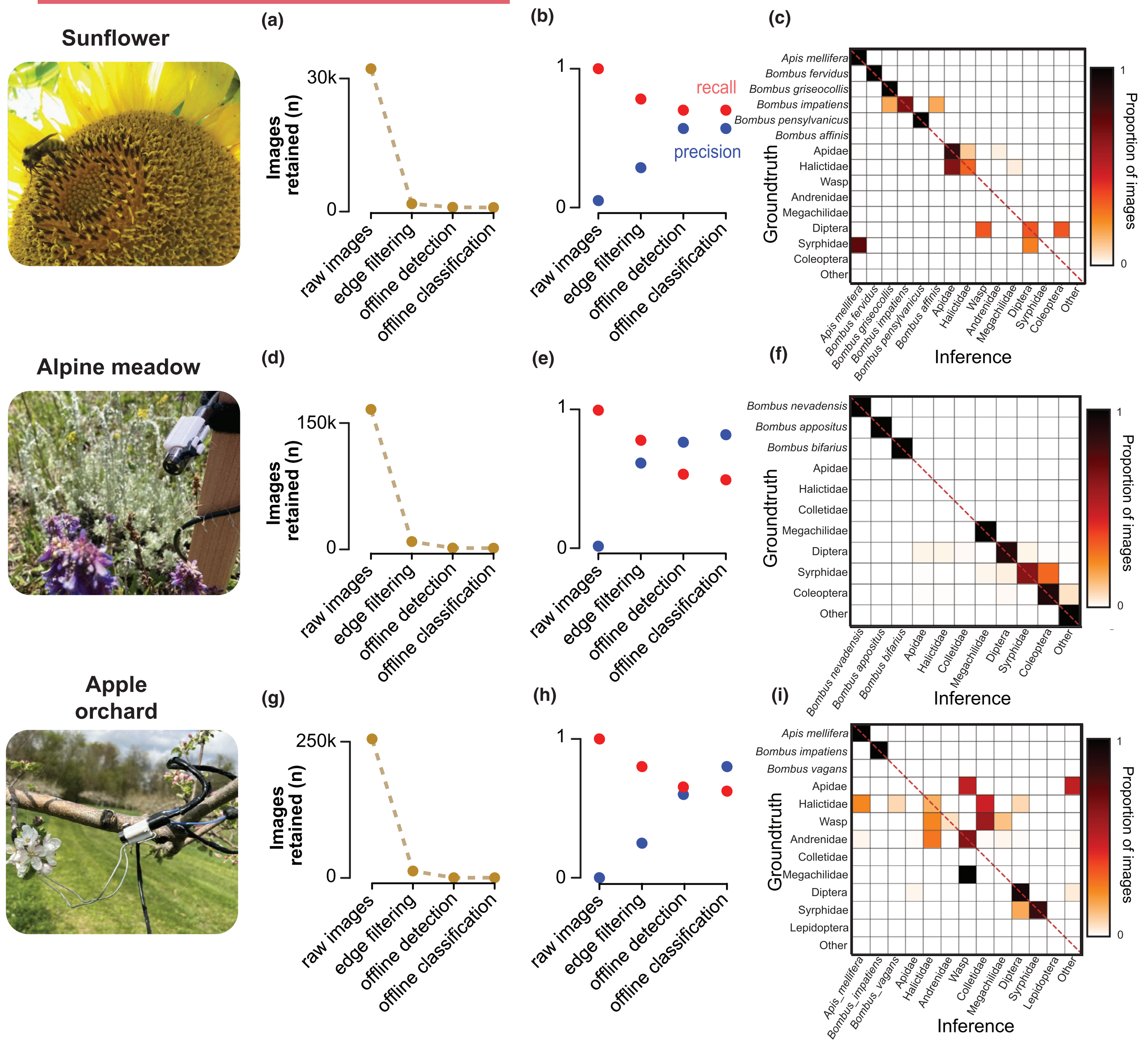

FIGURE 2 AutoPollS devices and workflow implemented across three field settings: Sunflowers within a diverse agriculture field at Arlington research station, WI (top row); an alpine meadow within Rocky Mountain Biological Laboratory, CO (middle row); and a commercial apple orchard in Dane County, WI (bottom row); (a, d, g) Total number of images saved from initial image acquisition to subsequent AutoPollS workflow steps outlined in Figure 1c: (1) image acquisition (2) on-device detection (3) offline detection (4) offline classification. (b, e, h) The precision (Equation 2) and recall (Equation 3) for each of the AutoPollS workflow steps, with Raw being equivalent to the initial unfiltered image acquisition. (c, f, i) Confusion matrices produced after all AutoPollS workflow (filtering) steps, displaying the ground truth labels on the *Y*-axis and inferred classification on the *X*-axis. The units represent the proportion of images normalized across the *x*-axis (ground truth observations).

has diverse potential applications, including automated quantification of ecosystem services (e.g. crop pollination services in apple) as well as non-lethal biodiversity monitoring. For example, AutoPollS correctly identified a federally endangered bumble bee (*Bombus affinis*, Figure 1d). Our results also demonstrate the value of integrating automated insect monitoring with environmental data to study insect response to dynamic environmental conditions, including microclimates.

However, our results also highlight remaining challenges in scalable, automated monitoring. First, for both detection and classification models, we saw reductions in performance from training data to our 'real-world' image data. Such reduced performance likely reflects the challenges of shifting domains from training data to real-world applications, including changes in image quality (such as resolution, framing, occlusion and blur) in our data sets compared to curated training image datasets. We also observed increased precision and decreased recall when threshold parameters optimized for one dataset (i.e. sunflower) were applied to different datasets (alpine meadow and apple).

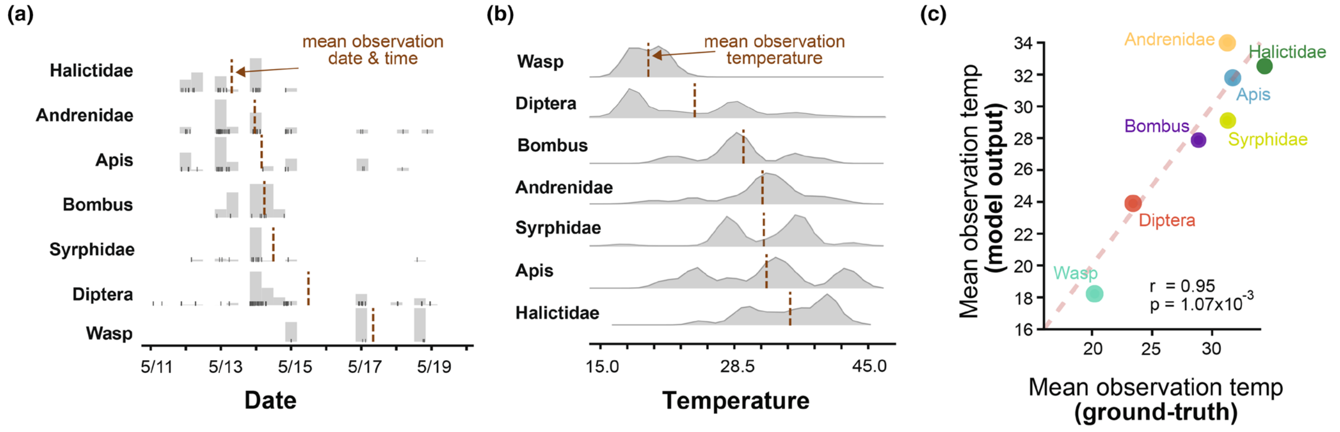

FIGURE 3 Analysis from the apple orchard deployment (a) Observed insect counts over time sorted by taxon, median plotted with dotted vertical line. (b) Kernel density estimates of observed visitation across temperature. (c) The comparison of the mean manually annotated temperature of visitation to the mean inferred temperature of visitation after AutoPollS workflow filtering steps (Pearson-$r=0.95$, $p=1.07\times10^{-3}$). The identity line is shown in dashed red.

While inclusion of even limited domain-specific (i.e. AutoPollS-generated) images here significantly improved model performance over fully 'naïve' models (Figure S10), further training on context-specific and diversified images that increasingly represent realistic image applications (including blur and lighting conditions, as well as sparse detections) could improve future model performance and easily integrate into the AutoPollS workflow. Accurate and reliable taxonomic classification could potentially be further improved by incorporating concurrent lines of evidence (such as geographic location) into model predictions. Incorporating this evidence may lead to more high-quality, research-grade, annotations and help address concerns in quality variability from crowd-sourced datasets. Beyond limitations of model performance, it remains unclear if the temporal sampling frequency achieved here (~0.2-1 Hz depending on configuration, Table S2) accurately captures flower visitation. Manual validation of AutoPollS with detailed manual observation to evaluate impacts of temporal sampling frequency (and camera orientation) on detection will be an important area for future work. Increased sampling frequency (e.g. via improved model efficiency or hardware implementations) could also facilitate other applications, such as quantification of on-flower behaviour.

Progressive integration of multiple model stages (edge and offline) also allows researchers to adjust for different use cases. For example, in studies where visitation by abundant species—but over extended time periods or spatial scales—is of primary interest, researchers may use higher (i.e. less permissive) confidence thresholds in the field to reduce data storage and improve the concentration of true positives in the resulting image data set. Such an approach may be particularly useful to quantify delivery of pollination services in crop systems where managed bees (e.g. *Apis mellifera* or *Bombus impatiens*) are likely to dominate. However, for applications where users want to avoid false negatives (e.g. to detect rare species), a lower (i.e. more permissive) confidence threshold will improve recall (and reduce false negatives), at the cost of larger image storage requirements. This integrated approach could facilitate feedback between data collection and model accuracy to address the challenge identified above; for example, AutoPollS can be deployed with a permissive threshold in novel ecosystems to generate image datasets for ecological research with minimal risk. Data generated from these field studies—after subsequent analysis and curation, using human-in-the-loop annotation and validation—can then be used to refine edge models, improving accuracy and reducing the necessity for data storage in the field, creating beneficial feedback between field applications and model development efforts. Looking forward, coordinating such efforts across labs and study systems will be critical for accelerating model improvement and ensuring that advances are broadly transferable.

Developing accurate and accessible approaches for robust insect monitoring across diverse environments, while of global importance, still faces critical methodological and technical challenges (Sittinger et al., 2024). Generalizable detection and classification models, for example, remain a significant challenge in computer vision (Tuia et al., 2022), with model performance in novel contexts suffering from biases (e.g. geographic or taxonomic) in training (Høye et al., 2021; Schneider et al., 2023; Tuia et al., 2022). AutoPollS adds to the growing toolkit for automated, deep learning-based pollinator monitoring (Ärje et al., 2020; Bjerge et al., 2022; Droissart et al., 2021; Geissmann et al., 2022; Høye et al., 2021; Tuia et al., 2022; van Klink et al., 2022) and may help address critical challenges of increasing the reliability and effectiveness of automated monitoring in diverse, real-world applications to support biodiversity conservation and ecosystem services.

AUTHOR CONTRIBUTIONS

Matthew A.-Y. Smith, Olivia M. Bernauer, Grace Melone, Brett Graham, Julia Wiessing, Rafael Salas, Jeremy Hemberger, Abhishek Bhattacharyya, Acacia T. S. Tang, Elena Rojas, Joshua San Miguel, Brian J. Spiesman, Claudio Gratton, James D. Crall conceived the ideas and designed methodology; Matthew A.-Y. Smith, Olivia M. Bernauer, Grace Melone, Julia Wiessing, Rafael Salas, Elena Rojas, James D. Crall collected the data; Matthew A.-Y. Smith, Olivia M.

Bernauer, Grace Melone, Julia Wiessing, Acacia T. S. Tang, Brian J. Spiesman, James D. Crall analysed the data; Matthew A.-Y. Smith, Olivia M. Bernauer, Grace Melone, Jeremy Hemberger, Brian J. Spiesman, Claudio Gratton, James D. Crall led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

ACKNOWLEDGEMENTS

We thank members of the Crall laboratory for helpful discussions and feedback on content, hardware design and manuscript revisions, Berry Brosi, Chris Anderson, Luca Pegoraro and Michael Dillon for testing and feedback on hardware and software design, Liz Griffith, Russell Groves and Scott Chapman for hosting and supporting on-farm research, John Mola for help in reviewing Bombus species identifications, as well as three anonymous reviewers and the editors at Methods in Ecology and Evolution. This research was supported by a Wisconsin Alumni Research Foundation Research Forward grant to JDC, CG and JSM, a Winslow Foundation grant to JDC, USDA NIFA AFRI grant no. 2022-67013-36275 from the USDA National Institute of Food and Agriculture to J.D.C.

CONFLICT OF INTEREST STATEMENT

The authors declare they have no competing interests.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210x.70304.

DATA AVAILABILITY STATEMENT

Data, Code, and the image dataset are available via https://doi.org/10.5281/zenodo.19360800 (Smith et al., 2026). The latest information and instructions on using the AutoPollS system are available via Github: https://github.com/crall-lab/autopolls or the AutoPollS website: https://autopolls.ai.

ORCID

Matthew A.-Y. Smith https://orcid.org/0000-0003-0913-1392
Olivia M. Bernauer https://orcid.org/0000-0002-4814-3188
Grace Melone https://orcid.org/0000-0002-5236-387X
James D. Crall https://orcid.org/0000-0002-8981-3782

REFERENCES

Ärje, J., Melvad, C., Jeppesen, M. R., Madsen, S. A., Raitoharju, J., Rasmussen, M. S., Iosifidis, A., Tirronen, V., Gabbouj, M., Meissner, K., & Høye, T. T. (2020). Automatic image-based identification and biomass estimation of invertebrates. *Methods in Ecology and Evolution*, *11*(8), 922–931.

Bjerge, K., Alison, J., Dyrmann, M., Frigaard, C. E., Mann, H. M. R., & Høye, T. T. (2023). Accurate detection and identification of insects from camera trap images with deep learning. *PLOS Sustainability and Transformation*, *2*(3), e0000051.

Bjerge, K., Karstoft, H., & Høye, T. T. (2025). Towards edge processing of images from insect camera traps. *Remote Sensing in Ecology and Conservation*, *11*, 573–589. https://doi.org/10.1002/rse2.70007

Bjerge, K., Mann, H. M. R., & Høye, T. T. (2022). Real-time insect tracking and monitoring with computer vision and deep learning. *Remote Sensing in Ecology and Conservation*, *8*(3), 315–327.

Corbet, S. A. (1990). Pollination and the weather. *Israel Journal of Botany*, *39*(1–2), 13–30.

Droissart, V., Azandi, L., Onguene, E. R., Savignac, M., Smith, T. B., & Deblauwe, V. (2021). PICT: A low-cost, modular, open-source camera trap system to study plant–insect interactions. *Methods in Ecology and Evolution/British Ecological Society*, *12*(8), 1389–1396.

Dwyer, B., Nelson, J., & Hansen, T. (2022). *Roboflow (Version 1.0) [Software]*. Roboflow. https://roboflow.com

Geissmann, Q., Abram, P. K., Wu, D., Haney, C. H., & Carrillo, J. (2022). Sticky Pi, a high-frequency smart trap to study insect circadian activity under natural conditions. *bioRxiv*. https://doi.org/10.1101/2021.08.11.455750

Gezon, Z. J., Wyman, E. S., Ascher, J. S., Inouye, D. W., & Irwin, R. E. (2015). The effect of repeated, lethal sampling on wild bee abundance and diversity. *Methods in Ecology and Evolution*, *6*(9), 1044–1054.

Høye, T. T., Ärje, J., Bjerge, K., Hansen, O. L. P., Iosifidis, A., Leese, F., Mann, H. M. R., Meissner, K., Melvad, C., & Raitoharju, J. (2021). Deep learning and computer vision will transform entomology. *Proceedings of the National Academy of Sciences of the United States of America*, *118*(2). https://doi.org/10.1073/pnas.2002545117

Høye, T. T., Montagna, M., Oteman, B., & Roy, D. B. (2025). Emerging technologies for pollinator monitoring. *Current Opinion in Insect Science*, *69*, 101367.

Huang, Z., Yang, S., Zhou, M., Gong, Z., Abusorrah, A., Lin, C., & Huang, Z. (2022). Making accurate object detection at the edge: Review and new approach. *Artificial Intelligence Review*, *55*(3), 2245–2274.

Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics. 10 January 2023.

Klaus, F., Ayasse, M., Classen, A., Dauber, J., Diekötter, T., Everaars, J., Fornoff, F., Greil, H., Hendriksma, H. P., Jütte, T., Klein, A. M., Krahner, A., Leonhardt, S. D., Lüken, D. J., Paxton, R. J., Schmid-Egger, C., Steffan-Dewenter, I., Thiele, J., Tscharntke, T., … Pistorius, J. (2024). Improving wild bee monitoring, sampling methods, and conservation. *Basic and Applied Ecology*, *75*, 2–11.

Klein, A.-M., Vaissière, B. E., Cane, J. H., Steffan-Dewenter, I., Cunningham, S. A., Kremen, C., & Tscharntke, T. (2007). Importance of pollinators in changing landscapes for world crops. *Proceedings of the Royal Society B: Biological Sciences*, *274*(1608), 303–313.

Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). Thresholding classifiers to maximize F1 score. *arXiv*. http://arxiv.org/abs/1402.1892

Olesen, J. M., Bascompte, J., Elberling, H., & Jordano, P. (2008). Temporal dynamics in a pollination network. *Ecology*, *89*(6), 1573–1582.

Ollerton, J. (2017). Pollinator diversity: Distribution, ecological function, and conservation. *Annual Review of Ecology, Evolution, and Systematics*, *48*(1), 353–376.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *arXiv*. http://arxiv.org/abs/1912.01703

Redmon, J., Divvala, S., Girshick, R. B., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. *Proceedings/CVPR, IEEE Computer Society conference on computer vision and pattern recognition* (pp. 779–788).

Reilly, J. R., Artz, D. R., Biddinger, D., Bobiwash, K., Boyle, N. K., Brittain, C., Brokaw, J., Campbell, J. W., Daniels, J., Elle, E., Ellis, J. D., Fleischer, S. J., Gibbs, J., Gillespie, R. L., Gundersen, K. B., Gut, L., Hoffman, G., Joshi, N., Lundin, O., … Winfree, R. (2020). Crop production in the USA is frequently limited by a lack of pollinators. *Proceedings of the Royal Society B: Biological Sciences*, *287*(1931), 20200922.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1–3), 157–173.

Schneider, S., Taylor, G. W., Kremer, S. C., & Fryxell, J. M. (2023). Getting the bugs out of AI: Advancing ecological research on arthropods through computer vision. *Ecology Letters*, *26*(7), 1247–1258.

Sittinger, M., Uhler, J., Pink, M., & Herz, A. (2024). Insect detect: An open-source DIY camera trap for automated insect monitoring. *PLoS One*, *19*(4), e0295474.

Smith, A.-Y. S., Bernauer, O. M., Melone, G., Graham, B., Wiessing, J., Salas, R., Hemberger, J., Bhattacharyya, A., Tang, T. S. A., Rojas, E., San Miguel, J., Spiesman, J. B., Gratton, C., & Crall, D. J. (2026). AutoPollS: A tool for automated monitoring of pollinators using deep learning [software, code, data]. *Methods in Ecology and Evolution*. https://doi.org/10.5281/zenodo.19360800

Spiesman, B. J., Gratton, C., Hatfield, R. G., Hsu, W. H., Jepsen, S., McCornack, B., Patel, K., & Wang, G. (2021). Assessing the potential for deep learning and computer vision to identify bumble bee species from images. *Scientific Reports*, *11*(1), 7580.

Ștefan, V., Workman, A., Cobain, J. C., Rakosy, D., & Knight, T. M. (2024). Utilising affordable smartphones and open-source time-lapse photography for monitoring pollinators. *bioRxiv*. https://doi.org/10.1101/2024.01.31.578173

Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training. *arXiv*. http://arxiv.org/abs/2104.00298

Tepedino, V. J., & Portman, Z. M. (2021). Intensive monitoring for bees in North America: Indispensable or improvident? *Insect Conservation and Diversity*, *14*(5), 535–542.

Tuia, D., Kellenberger, B., Beery, S., Costelloe, B. R., Zuffi, S., Risse, B., Mathis, A., Mathis, M. W., van Langevelde, F., Burghardt, T., Kays, R., Klinck, H., Wikelski, M., Couzin, I. D., van Horn, G., Crofoot, M. C., Stewart, C. V., & Berger-Wolf, T. (2022). Perspectives in machine learning for wildlife conservation. *Nature Communications*, *13*(1), 792.

van Klink, R., August, T., Bas, Y., Bodesheim, P., Bonn, A., Fossøy, F., Høye, T. T., Jongejans, E., Menz, M. H. M., Miraldo, A., Roslin, T., Roy, H. E., Ruczyński, I., Schigel, D., Schäffler, L., Sheard, J. K., Svenningsen, C., Tschan, G. F., Wäldchen, J., … Bowler, D. E. (2022). Emerging technologies revolutionise insect ecology and monitoring. *Trends in Ecology & Evolution*, *37*(10), 872–885.

Woodard, S. H., Federman, S., James, R. R., Danforth, B. N., Griswold, T. L., Inouye, D., McFrederick, Q. S., Morandin, L., Paul, D. L., Sellers, E., Strange, J. P., Vaughan, M., Williams, N. M., Branstetter, M. G., Burns, C. T., Cane, J., Cariveau, A. B., Cariveau, D. P., Childers, A., … Wehling, W. (2020). Towards a U.S. national program for monitoring native bees. *Biological Conservation*, *252*, 108821.

Yang, L. H., Postema, E. G., Hayes, T. E., Lippey, M. K., & MacArthur-Waltz, D. J. (2021). The complexity of global change and its effects on insects. *Current Opinion in Insect Science*, *47*, 90–102.

SUPPORTING INFORMATION

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

Figure S1. Overview of the AutoPollS software and edge inference workflow.

Figure S2. CPU inference speeds for Tensorflow-Lite models to process a single image running on the AutoPollS system. Red lines represent the median value across each model architecture.

Figure S3. Schematic representation of how the AutoPollS dataset is used across: (1) Object detection model training, validation, testing (2) classifier training, validation, testing (3) Confidence threshold tuning.

Figure S4. The number of images collected across the AutoPollS deployment in the apple orchard field site.

Figure S5. Images from the AutoPollS deployment in the apple orchard field site.

Figure S6. Annotated flower development stages for the field of view for each camera.

Figure S7. Temperatures recorded across the AutoPollS apple orchard deployment site.

Figure S8. F1-score parameter optimization search when performed separately for each AutoPollS deployment sites: (A) apple orchard, (B) sunflowers in a diverse agriculture field, (C) alpine meadow.

Figure S9. The estimated F1 and thresholds from a 2500 iteration bootstrap over the diversified sunflower field training data after a 5-fold cross validation split.

Figure S10. Comparing model performance across YOLOv11 architectures trained with (+AP) and without (no AP) images from the AutoPollS, which were supplemented into training, validation and testing.

Table S1. Estimates of the cost of an Autopolls unit with all features added and four cameras attached. All prices in USD estimates of prices around Spring of 2023.

Table S2. Runtimes for the Autopolls system as function of number of attached cameras using the 88 W/h battery.

Table S3. Full table of taxonomic classification from the AutoPollS classification model.

Table S4. Matching the AutoPollS classifier output labels to the labels used for AutoPollS analysis.

Table S6. Full table of taxonomic classification from the AutoPollS classification model without AutoPolls images.

How to cite this article: Smith, M.-Y., Bernauer, O. M., Melone, G., Graham, B., Wiessing, J., Salas, R., Hemberger, J., Bhattacharyya, A., Tang, A. T. S., Rojas, E., San Miguel, J., Spiesman, B. J., Gratton, C., & Crall, J. D. (2026). AutoPollS: A tool for automated monitoring of pollinators using deep learning. *Methods in Ecology and Evolution*, *00*, 1–11. https://doi.org/10.1111/2041-210x.70304