Towards Computer Vision and Deep Learning Facilitated Pollination Monitoring for Agriculture

Malika Nisal Ratnayake[1], Adrian G. Dyer[2,3], Alan Dorin[1]
[1] Faculty of Information Technology, Monash University, Australia
[2] School of Media and Communication, RMIT University, Australia
[3] Department of Physiology, Monash University, Australia
{malika.ratnayake, alan.dorin}@monash.edu, adrian.dyer@rmit.edu.au

Abstract

Globally, pollinators affect 35% of agricultural land and play a key role in food production. Consequently, monitoring is useful to understand the contribution insects make towards crop pollination. Traditional sampling techniques used in insect monitoring have several drawbacks, including that they are labour intensive and potentially unreliable. Some of these drawbacks may be overcome using computer vision and deep learning-based approaches to automate pollination monitoring. In this paper, we present a pipeline for computer vision-based pollination monitoring and propose a novel algorithm, Polytrack, that tracks multiple insects simultaneously in complex agricultural environments. Our algorithm uses deep learning and foreground/background segmentation to detect and track insects. We achieved precision and recall rates of 0.975 and 0.972 respectively when monitoring honeybees foraging in our test sites within the polytunnels of an industrial strawberry farm. Polytrack includes a flower identification module to automate collection of insect-flower interaction data, and a low-resolution processing mode that reduces computational demands placed on the processor to bring the software towards the requirements of low-powered monitoring hardware.

1. Introduction

Pollination is an integral requirement for food production and management of global ecosystems. Pollinators affect 35% of global agricultural land [15], supporting over 87 food crops worldwide [2]. The annual market value of animal pollination in global crop production is estimated to be around $235-577 billion USD [33]. Numerous insects including bees, flies, butterflies and ants contribute to crop pollination [33, 35]. However, insect species may vary in their pollination efficiency. In many instances, crop yield is directly correlated with pollinator population size [39] and improved pollinator habitat management can increase crop yield by as much as 25% [16]. Hence, understanding the pollinator communities of a particular crop and their effectiveness is essential in improving crop yield and the long term viability of a farming project [17].

Insect monitoring and sampling can be used to understand the role of pollination services of different species to a crop or an ecosystem. The economic benefits associated with pollination monitoring are estimated to exceed the costs incurred in implementing such services [9]. The importance of pollination monitoring has become more critical than ever because of increased insect population declines [34] caused by climate change [23] and other human activities [43, 29].

Direct observations and pan-traps are two widely used insect pollinator monitoring techniques [49]. Direct observation is an active method of surveying where observers conduct transect walks or observe a fixed area to count and identify insect visitors. Although direct observations are often straightforward to conduct, they may be unintentionally biased [14, 41] and the quality of observations may depend on the expertise of the observer [44]. Also, as the sources of observations are not preserved, the accuracy of data may later be questioned [13], especially because human attention can be poor for non-foveal visual tasks and moving objects [20].

Pan-traps are a passive method of insect sampling where painted water-filled bowls are used to trap insects. Compared to direct observations, pan-traps can sample more insect species [49] and are better suited to taxonomic categorisation since the insects themselves are retained for later examination. But results of pan-trapping may be biased as some insect species are unlikely to get caught [10], and pan traps provide no functional evidence that an insect does visit or pollinate flowers.

Researchers and agriculturists use data from pan-trapping and observations to predict the value of insect species to crop pollination. However, these sampling techniques can only be used to collect data related to insect abundance, not pollination behaviour. This might prompt researches to incorrectly identify plant visitors as pollinators [28]. Computer vision and deep learning-based techniques can be used to overcome these drawbacks with traditional surveying techniques by analysing insect behaviour and maintaining event and interaction records [13, 30, 21].

Computer vision can be used to monitor pollination by tracking insect pollinators in high spatiotemporal resolution [21]. Previously, computer vision-based tracking programs have been developed to research the behavioural traits of insects [8, 32, 47]. However, their application is often confined to laboratories and controlled environments, and many require human intervention [19]. Tracking insect pollinators in natural environments is complex because these are highly dynamic and subject to changes in illumination and movements caused by wind and animals. The insect behaviour itself may further complicate tracking as they may leave or enter a video frame arbitrarily or crawl behind leaves. To be of practical value a computer vision-based pollination monitoring system should be robust enough to track insects through the aforementioned complexities without human intervention.

Previous research has proposed a Hybrid Detection and Tracking (HyDaT) algorithm [36] to track individual honeybees foraging among wildflowers. HyDaT tracks one insect at a time and requires preprocessing of videos to track multiple insects. This makes it unsuitable for pollination monitoring as often multiple insects forage in an area simultaneously, and preprocessing hours of videos can be laborious and prohibits time-sensitive real world interventions to modulate and improve pollination outcomes.

In the current work, we extend the methods of HyDaT and propose a novel algorithm "Polytrack" capable of tracking multiple insects simultaneously, to facilitate automation.

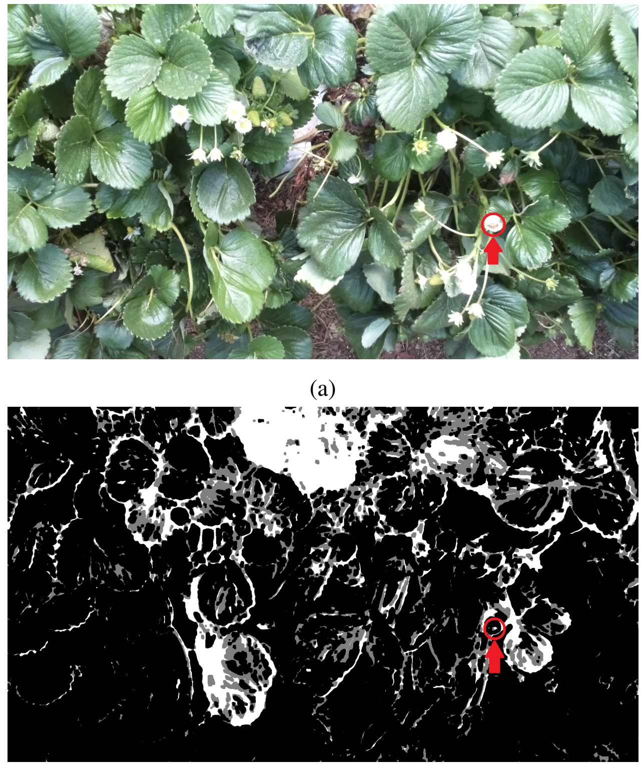
The primary contributions of this work are:

- A computer vision-based pollination monitoring pipeline.
- An algorithm (Polytrack) to track multiple insects simultaneously in agricultural setups.
- A flower identification module and low-resolution processing mode.
- Pretrained YOLOv4 model and Polytrack software [1].

2. Related Work

In this section, we summarise computer vision-based insect tracking focusing on suitability for pollinator monitoring. Tracking insects consists of two main components, detecting the position of the insect in a frame, and building a coherent trajectory by linking positional data across frames [13].

[1] https://github.com/malikaratnayake/Polytrack_v1



(a)

(b)

Figure 1: **Difficulty associated with detecting the position of an insect (honeybee) in a windy environment.** (a) RGB image of the video frame. (b) Foreground masks obtained using foreground/background segmentation (KNN background subtractor [53]). Pixels belonging to moving objects are indicated in white. Position of the insect is marked with a red circle.

Previous algorithms have used invasive and non-invasive methods to detect and track insects. Invasive methods that mark insects with tags are unsuitable for pollination monitoring as tagging insects is laborious and unrealistic, in agricultural setups. Therefore, non-invasive techniques with unmarked insects are more suitable for pollinator monitoring.

Segmentation methods such as foreground/background (FG/BG) segmentation and thresholding are widely used in unmarked insect tracking to identify the position of insects in a video frame [32, 8, 26, 50, 31, 37, 47]. FG/BG segmentation or colour thresholding techniques are efficient in controlled environments where background and illumination are constant and a significant contrast exists between objects and their background [13]. However, pollinator monitoring in semi-controlled or uncontrolled dynamic environments makes the application of background-subtraction for pollinator monitoring highly challenging (Figure 1).

Deep learning facilitated detection and tracking techniques help to overcome drawbacks associated with FG/BG segmentation techniques. Convolutional Neural Networks (CNN) can be used in taxonomic identification [27, 42] and detection of insects in individual frames irrespective of inter-frame changes and complex patterns in the frame[38, 4, 42, 6, 22, 12]. However, deep learning-based techniques require training with a large dataset to work efficiently [5] and require relatively high computational resources.

The drawbacks of the above detection methods are overcome to some extent by the hybrid detection algorithm, HyDaT [36]. It uses both FG/BG segmentation and deep learning-based models for detection and intelligently switches between the two detection models based on the variations in the background of the video. The modular design of HyDaT enables the use of different detection models. In the current work, we adopt the main working principles of HyDaT algorithm to track multiple insects simultaneously in an agricultural environment.

3. Methods

3.1. Computer Vision-based Pollination Monitoring

In this section, we propose a pipeline for computer vision-based pollination monitoring that consists of three stages (Figure 2). The initial stage is for data acquisition where cameras or IoT-based devices such as a Raspberry Pi are deployed to record videos of pollinators. In the data extraction stage, recorded videos are processed using tracking algorithms such as Polytrack to extract pollinator movements and flower positions. In the third stage of the pipeline, extracted tracks are analysed to draw conclusions on pollination levels of flowers, pollinator abundance and pollination efficiency of insects.

3.2. Polytrack Algorithm

Polytrack is designed to extract pollination related data from video recordings. These data include the position of flowers and insect movement trajectories. To enable efficient pollination monitoring Polytrack is equipped with: (i) a flower identification component to analyse video frames and detect fully open flowers; (ii) a detection and tracking component to track pollinators from their first appearance in the video to their exit; (iii) a low-resolution mode to accelerate video processing when there are no insects in the camera view. An overview of Polytrack is shown in Figure 3. The rest of this section presents each component.

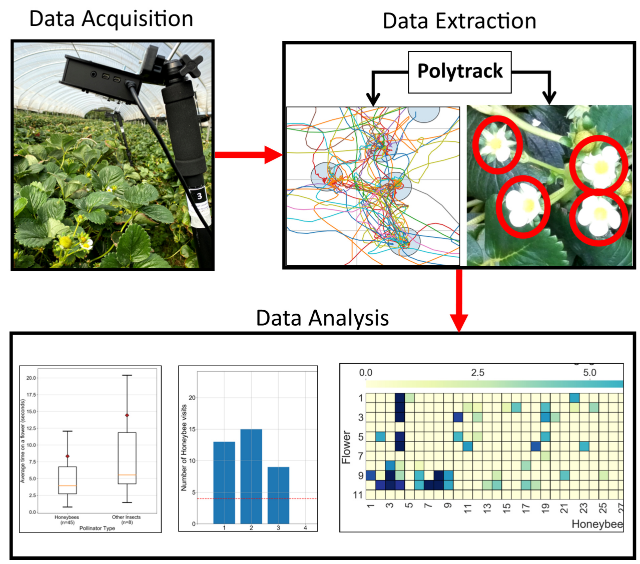


Figure 2: **Proposed pipeline for computer vision-based pollination monitoring.**

3.2.1 Flower Identification

Locating the positions of flowers is important when monitoring pollinators and assessing their efficiency. At the start of each video sequence, Polytrack uses the deep learning-based detection model (Section 3.2.2) to identify the position of flowers in the frame. The position and area of each fully open flower is recorded for later analysis.

3.2.2 Detection and Tracking

We extend the methods presented in HyDaT [36] to track multiple insects simultaneously from their first appearance in the video to their exit. Like HyDaT, Polytrack is designed with a modular architecture, where any detection model can be used based on the application and current state of the art. For the current implementation, we used YOLOv4 as the deep learning-based detection model and K-nearest neighbours-based (KNN) segmentation algorithm [53] as the foreground/background segmentation-based detection model.

Deep learning-based detection model. We use a Convolutional Neural Network (CNN)-based YOLOv4 object detection algorithm [3] for its speed and accuracy [3]. The use of a deep learning model enables important taxonomic identification of specific insects, enabling species-wise pollination evaluations. In the current implementation, YOLOv4 model will be used to identify a specific insect species. Details on the dataset preparation and training are provided in Section 4.1.

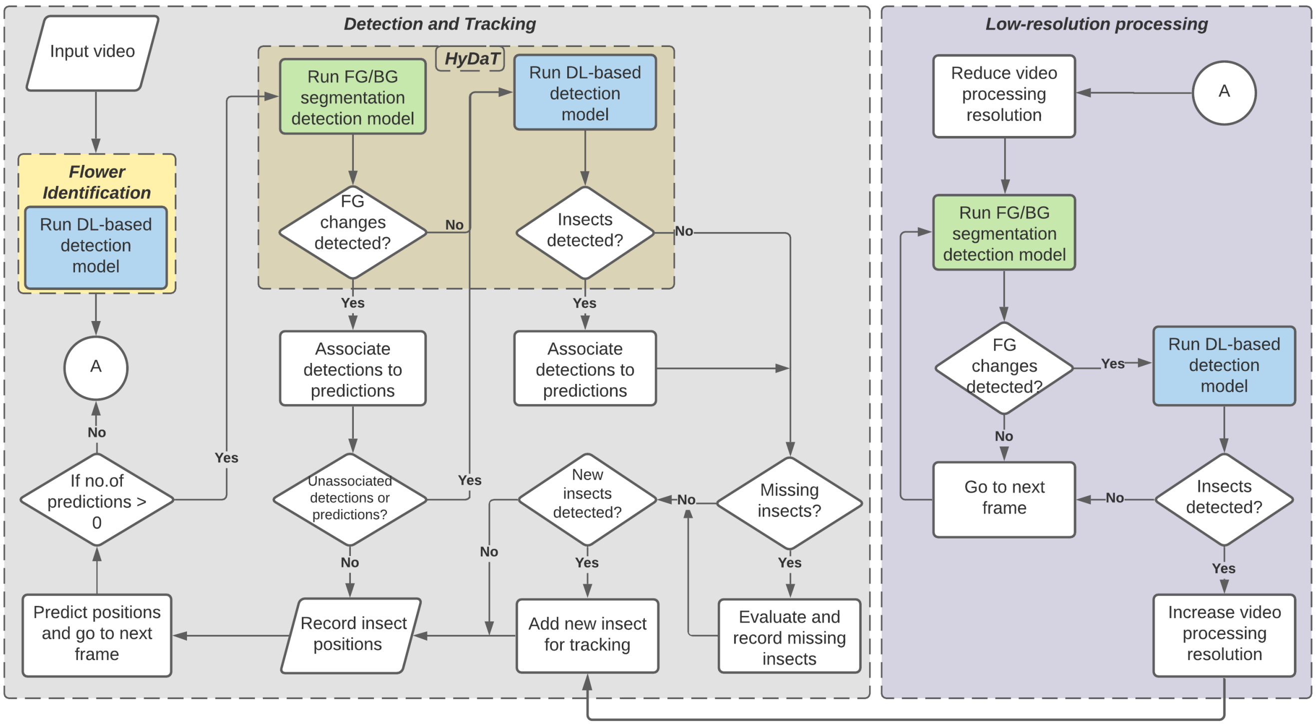


Figure 3: **Overview of the proposed Polytrack algorithm.** Methods adopted from HyDaT [36] are highlighted in brown.

Foreground/background (FG/BG) segmentation-based detection model. We use K-nearest neighbours (KNN)-based foreground/background segmentation [53] (OpenCV 3.4.1 [7]) to detect foreground changes in the video. The resulting binary image is passed through a median filter and an erosion-based morphological filter to remove noise. Next, contours of the foreground detections are extracted from the binary image and filtered based on their enclosing area to remove areas of movement less than a predetermined minimum pixel count covered by the focal insect.

At the start of each video sequence, Polytrack uses the deep learning-based detection model to identify and record the position of flowers in the frame. After the flower identification, the algorithm processes the video in low-resolution (Section 4.3) until an insect is detected.

When an insect appears in the frame, the deep learning-based model detects its position and then identifies the species. If multiple insects are detected simultaneously, the algorithm evaluates the accuracy of each respective detection and identification based on the associated confidence of detection and the distance between each. This approach ensures individual insect tracking with a low probability of false positives. When an insect is detected, Polytrack switches to full-resolution processing to enhance accuracy. After the initial detection of an insect, if there are a low number of regions of inter-frame change apparent within the frame, FG/BG segmentation is used to identify the position of the insect in subsequent frames, otherwise deep-learning is used to minimise false-positive detections. Whenever FG/BG segmentation is unable to locate the position of an insect, deep learning is used for individual detection.

When multiple insects are being tracked simultaneously, Polytrack first utilises FG/BG segmentation to identify their positions. If the positions of one or more insects are not detected using this model, deep learning-based detection will be used to detect the remaining insects. If any detections cannot be associated with a given track, the deep learning model is used to identify the possible appearance of a new insect.

Missing insects. When both detection models fail to detect the position of an insect being tracked, it will be considered a "missing" insect. If the last detected position of the insect is at the edge of the frame and the predicted position for the current frame is out of the frame border, the track is terminated assuming the insect to have left the frame. However, if the insect last appeared in the middle of the frame, the algorithm waits for the insect to be re-detected within a predefined radius from its predicted position. (This models predictive behaviour that helps allow human brains to solve occlusion problems [46].) If the insect is not detected within a predefined number of frames, the track is terminated.

Data association. Polytrack is designed to track multiple insects simultaneously from their first appearance in the video until they exit the camera's view. A "predict and detect" approach based on a constant velocity model is used to calculate an insect's track over successive frames. This method was considered over the Kalman filter [48] approach due to the high variability in insect movement across successive frames. Using the constant velocity model, in a set of three successive frames, the predicted position of the insect in the third is calculated from the detected positions in the first two frames, assuming constant insect velocity over the three frames [51, 25]. The predicted position P_k of the insect in frame k of the video is defined as:

$$P_k = [x_pk, y_pk]^T = A * [D_{k-1}, D_{k-2}]^T \tag{1}$$

where,

$$A = \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 \end{bmatrix}$$

In equation 1, x_pk and y_pk refer to coordinates of the predicted position of the insect in frame k and $[D_{k-1}, D_{k-2}]$ are the detected positions of the insect in the two previous frames.

When an insect is first detected, the predicted position for the next frame is assumed to be the same as its current position (as there are no preceding frames). We used the Hungarian algorithm [24] to associate predicted positions of insects with detections to form a track. The Hungarian algorithm requires a square cost matrix for accurate association. However, due to missing insects and false-positive detections caused by changes in the environment, the resulting cost matrix might not be square. In such instances, the cost matrix of the Hungarian algorithm is padded with zeros to construct a square matrix.

3.2.3 Low-resolution processing

Video sequences recorded in agricultural setups may contain extended periods where no pollinator is present within the camera field of view. Processing videos in high-quality during these periods wastes computer resources. As a solution, we introduced a low-resolution processing mode that processes video in low-resolution when no insects are being tracked. (This mirrors how the primate visual system efficiently solves detection and tracking tasks [52].)

In low-resolution processing mode, the resolution of the original video is reduced and processed using the FG/BG segmentation-based detection model. If there are significant changes in the foreground, corresponding pixel blobs will be analysed based on their area. The minimum area covered by an insect in the low-resolution mode ($area_{LR}$) is defined as follows.

$$area_{LR} = \left[\frac{w_{LR} \times h_{LR}}{w_{FR} \times h_{FR}}\right] \times area_{FR} \tag{2}$$

where, w_{LR} and h_{LR} are the frame width and height in the low-resolution mode, w_{FR} and h_{FR} are the frame width and height of the original video and $area_{FR}$ is the minimum area covered by the insect in the original video. If the FG/BG segmentation-based model detects blobs of moving with an area greater than $area_{LR}$, frames will be processed using the deep learning-based detection model to identify insects. Upon detection of an insect, Polytrack switches to full-resolution processing to track it.

4. Experimental Evaluation

In this section, we present implementation details (Section 4.1), compare the performance of our algorithms against state-of-the-art methods and ground observations to evaluate its tracking accuracy (Section 4.2) and the performance of the low-resolution mode (Section 4.3). Finally, we present an example data analysis to demonstrate the capabilities of our methods in pollination monitoring (Section 4.4).

4.1. Implementation details

Data acquisition. Data required for the study was recorded at Sunny Ridge strawberry farm, Victoria, Australia in March 2020. Honeybees (*Apis mellifera*) were used as the study subject since they are a pollinator of strawberry [17] and are used around the world as managed pollinators. 49 videos, each five minutes long, totalling up to four hours in length, were recorded between the hours of 10am and 2pm in a strawberry polytunnel at the farm. We used a Raspberry Pi Camera v2 with video resolution 1920×1080, 30 fps. The camera was set approximately 70 cm above the strawberry flowers. Dimensions of the area covered by the camera were $600mm \times 340mm$ (width of a planted hydroponic strawberry row), the average area covered by a honeybee at this range was 1001 ± 475.3 pixels.

Dataset preparation and training the YOLOv4 model. We created a custom dataset of 411 images with 456 honeybee instances and 87 strawberry flowers. The prepared dataset was annotated with bounding boxes using the Computer Vision Annotation Tool [40]. The YOLOv4 model was then trained using Tensorflow [1] with a learning rate of 0.001. The Mean Average Precision (mAP) of the trained model was 94.9%.

Development of software. The software was developed in Python (3.7.0) with Computer Vision Library (OpenCV) 3.4.2 and Tensorflow 2.3.0. Experiments were conducted in MASSIVE high computing infrastructure [18] with Intel

Xeon Gold 6150 (2.70 GHz) CPU, 55 GB RAM, NVIDIA Tesla P4 GPU and CentOS Linux (7). Data analysis was conducted using NumPy 1.16.2, Pandas 0.24.2 and Matplotlib 3.0.3.

4.2. Detection rate and tracking accuracy

The detection rate and tracking accuracy of the Polytrack algorithm were evaluated for 5 test videos by using precision (Equation 3) and recall (Equation 4) rates as evaluation matrices.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (3)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (4)$$

where $TruePositive$ is the total number of correct detections in all frames; $FalseNegative$ is the total number of undetected honeybees in frames and $FalsePositive$ is the total number of incorrectly detected positions of honeybees. Identity swaps in tracks were recorded as $FalsePositives$. Test videos comprised of instances where no honeybees, one honeybee or multiple honeybees were present simultaneously in a video frame. All videos contained natural variations in background and foreground illumination and foliage movement (Figure 4).

We compared the performance of the Polytrack algorithm against HyDaT [36] and stand-alone YOLOv4 [3] detection. We found it meaningless to compare our results with other existing insect tracking software as they were designed to track insects in controlled environments and failed to differentiate background movement from insect movement during pilot studies. Table 1 presents details of test videos and tracking results.

In our test videos, Polytrack was able to track honeybees with an overall precision of 0.975 and a recall of 0.972. Compared to HyDaT and YOLOv4 model, Polytrack achieved higher recall rates for four test videos and the overall increase in recall value was 18% and 14% respectively, which was an 84% and an 81% relative reduction in error compared to HyDaT and YOLOv4 respectively. Polytrack uses a combination of deep learning and FG/BG segmentation-based detection models to identify insect position. This enables Polytrack to be used with a deep learning model trained with a small training dataset and still achieve higher recall compared to stand-alone deep learning-based detection. YOLOv4 detected insect positions with higher precision for 3 test videos. However, the overall precision of YOLOv4 was 0.846 due to identity swaps. Polytrack obtained an overall precision score of 0.975, which was a 15% improvement (84% relative reduction in error) compared to the second best stand-alone

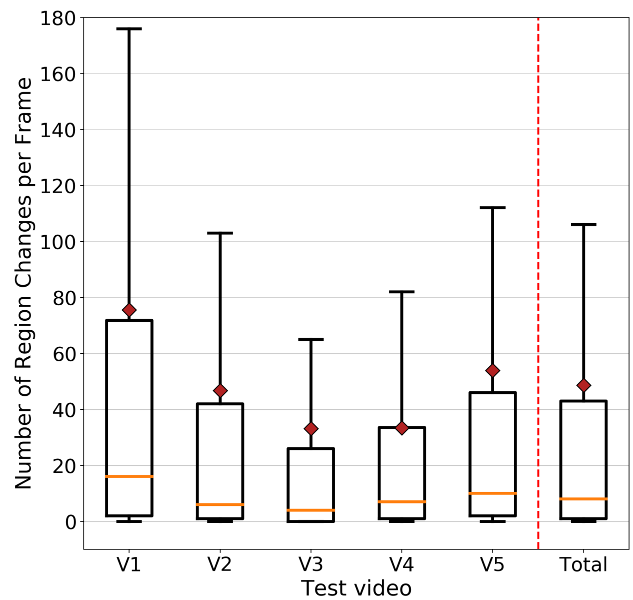


Figure 4: **Box plot showing the distribution of the number of image region changes per frame in test videos.** The number of image regions is the number of non-intersecting regions of area greater than that of the honeybee. The red diamond indicates the mean value and orange line shows the median.

YOLOv4. In most cases, Polytrack was able to track insects from their first appearance in the frame to the exit without breaking the track and while maintaining their identities. The HyDat algorithm partially or completely missed tracking when multiple honeybees were present inside the frame.

4.3. Low-resolution mode

We evaluated the effectiveness of the low-resolution processing mode to increase tracking speed for five test videos. All videos consisted of periods where there was no honeybee present inside the frame. We compared the total processing time and average processing speed (frames per second) with and without using low-resolution processing mode. A resolution of 852×480 was used in low-resolution mode to process videos. Results are shown in Table 2. The low-resolution mode reduced total processing time by 61.2% and increased processing speed by a factor of 2.58 (258%) in our sample videos.

4.4. Example data analysis

In this section we present an example data analysis demonstrating the application of our methods for pollination monitoring. We used Polytrack to extract flower positions and movement trajectories of honeybees from the

Table 1: **A quantitative comparison of the Polytrack algorithm's tracking performance against HyDaT [36] and standalone YOLOv4 [3].** Algorithm performance is assessed using precision and recall matrices. # HBs in video denotes total number of honeybees recorded in the video sequence and the maximum number of honeybees recorded simultaneously is presented in brackets. The best performing algorithm is presented in bold.

Video	Total frames	# frames with HBs	# HBs in video	No. of tracks generated			Precision			Recall		
				Polyt.	HyDaT	YOLO	Polyt.	HyDaT	YOLO	Polyt.	HyDaT	YOLO
V1	8078	650	2 (1)	2	2	4 [1]	0.995	0.992	**1.000**	0.980	**1.000**	0.784
V2	7541	398	2 (2)	2	2	3 [1]	0.997	0.994	**1.000**	**1.000**	0.884	0.857
V3	7895	564	5 (1)	6 [2]	6 [2]	24 [1,2]	0.857	0.793	**0.882**	**0.986**	0.890	0.656
V4	8183	2106	3 (2)	3	4 [1]	3	**0.999**	0.997	**0.999**	**0.996**	0.961	0.934
V5	8212	1573	6 (3)	8 [3]	3 [4]	8 [3]	**0.977**	0.440	0.497	**0.925**	0.447	0.823
Overall	39909	5291	18	21	17	42	**0.975**	0.831	0.846	**0.972**	0.826	0.852

[1] Multiple tracklets generated by a single honeybee.
[2] New track generated by false-positive detections.
[3] Honeybee(s) occluded from the view generated multiple tracklets.
[4] Completely missed honeybees.

Table 2: **An analysis of processing speed of the Polytrack algorithm with and without the low-resolution processing mode.** Processing time denotes the time taken by the algorithm to process a video and the processing speed shows the average number of frames processed in a second.

Video	Processing Time (sec)		Processing Speed (fps)	
	With Low-Res	Without Low-Res	With Low-Res	Without Low-Res
V1	**283.08**	797.35	**28.54**	10.13
V2	**225.31**	701.56	**33.47**	10.75
V3	**272.85**	748.41	**28.94**	10.55
V4	**366.21**	754.85	**22.35**	10.84
V5	**343.79**	839.36	**23.89**	9.78
Overall	**1491.24**	3841.53	**26.76**	10.39

dataset (Section 4.1). Figure 5a shows the field of view of the camera and position of flowers automatically extracted. Polytrack processed 4 hours of video at 31.12 fps and extracted 85 honeybee tracks (Figure 5b).

In strawberry, cross-pollination leads to a higher quality fruit set [45]. To encourage cross-pollination, different varieties of strawberries are planted in adjacent rows. The level of cross-pollination can be estimated through the general pollen flow direction, which is usually dictated by the pollinator movement direction. Figure 5b shows a map of honeybee trajectories extracted using Polytrack and calculations of general pollen flow direction.

The number of visits to a flower is an important indicator of its pollination level. Research shows that strawberry flowers required at least 4 visits from pollinators to fully fertilise a flower [17, 11]. Similarly, time spent by pollinators on flowers increases the level of pollen dispersal, and the quality of pollination [11]. Figures 5c and 5d show the number of honeybee visits each flower in the field of view of the camera received and the distribution of time spent by honeybees on each flower during the study period.

5. Future Work

We identify several improvements to our methodology for future work. Currently, Polytrack requires several predefined parameters to facilitate the tracking process. Automating parameter selection will enable wide use of our methods with minimum setup. Also, we plan to develop a lighter version of the algorithm which can run on IoT devices such as Raspberry Pi and NVIDIA Jetson. This would enable real-time processing of video data, giving access to instantaneous pollination information. The current implementation works with a fixed camera setup. However most commercial agriculture extends over large areas and monitoring pollinators using fixed cameras would not be practical or cost-effective. Therefore, as future work, we plan to extend our methods to monitor pollinators with moving cameras. Furthermore, the current implementation of the algorithm works well in crops with flowers arranged in essentially a two-dimensional carpet. Nevertheless, some crops are more three dimensional in structure, and it would be beneficial for future work to extend the methods to accommodate three-dimensional tracking.

6. Conclusions

With the increase in global food demand and the decline of pollinator populations, it has become very important to monitor and manage insect pollinators carefully to maximise food production. Although currently, traditional

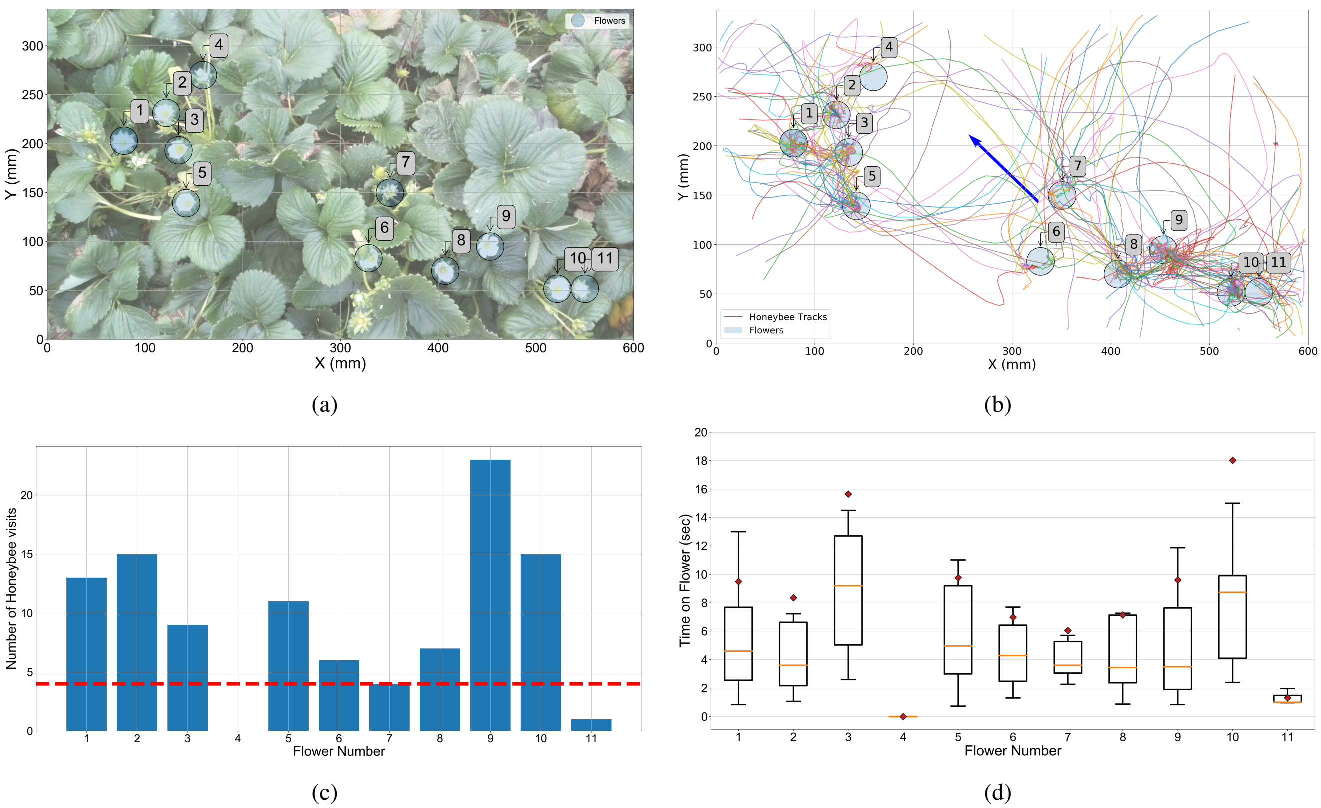


Figure 5: **Pollination monitoring analysis conducted on trajectories extracted using Polytrack.** (a) The field of view of the camera and the flowers identified through the algorithm, (b) Extracted honeybee trajectories. The blue arrow shows the general direction of the pollen flow calculated as the mean direction of travel vector for all honeybee travel segments in the sequence, (c) the Number of visits made by honeybees to each flower in the video frame. Redline shows the minimum number of visits required to fertilise a strawberry flower [17, 11], (d) The box plot shows the distribution of time honeybees spent foraging on flowers. The red diamond indicates the mean value and orange line shows the median.

insect sampling techniques such as pan-traps and visual observations are used for this purpose, there are several drawbacks associated with them. The use of computer vision techniques in pollinator monitoring has the potential to overcome most of these drawbacks. In this research, we propose a pipeline for computer vision-based pollination monitoring. To facilitate extracting pollination related data from videos we present a novel pollinator monitoring algorithm, Polytrack, capable of tracking multiple insect pollinators simultaneously in complex agricultural setups. Polytrack consist of components to (i) identify the position of flowers in a frame; (ii) detect and track insect pollinators; (iii) accelerate the tracking process. Our algorithm achieved a precision rate of 0.975 and a recall rate of 0.972 in our experiments, which is an improvement over the available alternatives. We demonstrate the use of our algorithm by analysing the pollination behaviour of honeybees foraging in a strawberry polytunnel. We calculated pollination related matrices such as pollen flow direction, flower visits and foraging time on flowers, which would have been infeasible with other pollination monitoring techniques.

Acknowledgements

The authors would like to thank Sunny Ridge Australia for the opportunity to conduct research at their farm, and Dr. Scarlett R. Howard for assisting in data collection. Authors were supported by the Australian Research Council Discovery Projects grant DP160100161 and Monash-Bosch AgTech Launchpad primer grant. AD acknowledges AgriFutures grant PRJ-012993.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016. 5

[2] Marcelo A. Aizen, Lucas A. Garibaldi, Saul A. Cunningham, and Alexandra M. Klein. How much does agriculture depend on pollinators? Lessons from long-term trends in crop production. *Annals of Botany*, 103(9):1579–1588, 2009. 1

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3, 6, 7

[4] Franziska Boenisch, Benjamin Rosemann, Benjamin Wild, David Dormagen, Fernando Wario, and Tim Landgraf. Tracking all members of a honey bee colony over their lifetime using learned models of correspondence. *Frontiers in Robotics and AI*, 5:35, 2018. 3

[5] Connor Bowley, Alicia Andes, Susan Ellis-Felege, and Travis Desell. Detecting wildlife in uncontrolled outdoor video using convolutional neural networks. In *2016 IEEE 12th International Conference on e-Science (e-Science)*, pages 251–259. IEEE, 2016. 3

[6] Katarzyna Bozek, Laetitia Hebert, Alexander S Mikheyev, and Greg J Stephens. Towards dense object tracking in a 2d honeybee hive. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4185–4193, 2018. 3

[7] G Bradski. The OpenCV Library. *Dr Dobbs Journal of Software Tools*, 25:120–125, 2000. 4

[8] Kristin Branson, Alice A Robie, John Bender, Pietro Perona, and Michael H Dickinson. High-throughput ethomics in large groups of drosophila. *Nature methods*, 6(6):451–457, 2009. 2

[9] Tom D Breeze, Alison P Bailey, Kelvin G Balcombe, Tom Brereton, Richard Comont, Mike Edwards, Michael P Garratt, Martin Harvey, Cathy Hawes, Nick Isaac, et al. Pollinator monitoring more than pays for itself. *Journal of Applied Ecology*, 58(1):44–57, 2021. 1

[10] James H Cane, Robert L Minckley, and Linda J Kervin. Sampling bees (hymenoptera: Apiformes) for pollinator community studies: pitfalls of pan-trapping. *Journal of the Kansas entomological society*, pages 225–231, 2000. 1

[11] M Chagnon, J Gingras, and D De Oliveira. Effect of honey bee (hymenoptera: Apidae) visits on the pollination rate of strawberries. *Journal of Economic Entomology*, 82(5):1350–1353, 1989. 7, 8

[12] Marta de Oliveira Barreiros, Diego de Oliveira Dantas, Luís Claudio de Oliveira Silva, Sidarta Ribeiro, and Allan Kardec Barros. Zebrafish tracking using yolov2 and kalman filter. *Scientific reports*, 11(1):1–14, 2021. 3

[13] Anthony I Dell, John A Bender, Kristin Branson, Iain D Couzin, Gonzalo G de Polavieja, Lucas PJJ Noldus, Alfonso Pérez-Escudero, Pietro Perona, Andrew D Straw, Martin Wikelski, et al. Automated image-based tracking and its application in ecology. *Trends in ecology & evolution*, 29(7):417–428, 2014. 1, 2

[14] RLH Dennis, TG Shreeve, NJB Isaac, DB Roy, PB Hardy, R Fox, and J Asher. The effects of visual apparency on bias in butterfly recording and monitoring. *Biological conservation*, 128(4):486–492, 2006. 1

[15] FAO. Why bees matter; the importance of bees and other pollinators for food and agriculture. *FAO*, 2018. 1

[16] Lucas A Garibaldi, Fabrice Requier, Orianne Rollin, and Georg K S Andersson. Towards an integrated species and habitat management of crop pollination. *Current Opinion in Insect Science*, 21:105–114, 2017. 1

[17] Lucas A Garibaldi, Agustín Sáez, Marcelo A Aizen, Thijs Fijen, and Ignasi Bartomeus. Crop pollination management needs flower-visitor monitoring and target values. *Journal of Applied Ecology*, 57(4):664–670, 2020. 1, 5, 7, 8

[18] Wojtek James Goscinski, Paul McIntosh, Ulrich Claus Felzmann, Anton Maksimenko, Christopher John Hall, Timur Gureyev, Darren Thompson, Andrew Janke, Graham Galloway, Neil EB Killeen, et al. The multi-modal australian sciences imaging and visualization environment (massive) high performance computing infrastructure: applications in neuroscience and neuroinformatics research. *Frontiers in Neuroinformatics*, 8:30, 2014. 5

[19] Lars Haalck, Michael Mangan, Barbara Webb, and Benjamin Risse. Towards image-based animal tracking in natural environments using a freely moving camera. *Journal of neuroscience methods*, 330:108455, 2020. 2

[20] Yoriko Hirose, Alan Kennedy, and Benjamin W Tatler. Perception and memory across viewpoint changes in moving images. *Journal of Vision*, 10(4):2–2, 2010. 1

[21] Toke T Høye, Johanna Ärje, Kim Bjerge, Oskar LP Hansen, Alexandros Iosifidis, Florian Leese, Hjalte MR Mann, Kristian Meissner, Claus Melvad, and Jenni Raitoharju. Deep learning and computer vision will transform entomology. *Proceedings of the National Academy of Sciences*, 118(2), 2021. 2

[22] Mat Kelcey. Counting bees on a Raspberry Pi with Convolutional Network, 2018. http://matpalm.com/blog/counting_bees/. 3

[23] Jeremy T Kerr, Alana Pindar, Paul Galpern, Laurence Packer, Simon G Potts, Stuart M Roberts, Pierre Rasmont, Oliver Schweiger, Sheila R Colla, Leif L Richardson, et al. Climate change impacts on bumblebees converge across continents. *Science*, 349(6244):177–180, 2015. 1

[24] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[25] Baptiste Magnier, Gaëtan Ekszterowicz, Joseph Laurent, Matthias Rival, and François Pfister. Bee hive traffic monitoring by tracking bee flight paths. In *13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, January 27-29, 2018, in Funchal, Madeira, Portugal*, pages 563–571, 2018. 5

[26] Hauke Jürgen Mönck, Andreas Jörg, Tobias von Falkenhausen, Julian Tanke, Benjamin Wild, David Dormagen, Jonas Piotrowski, Claudia Winklmayr, David Bierbach, and Tim Landgraf. Biotracker: an open-source computer vision framework for visual animal tracking. *arXiv preprint arXiv:1803.07985*, 2018. 2

[27] Rupesh G Mundada and VV Gohokar. Detection and classification of pests in greenhouse using image processing. *IOSR Journal of Electronics and Communication Engineering*, 5(6):57–63, 2013. 3

[28] Rory S O'Connor, William E Kunin, Michael PD Garratt, Simon G Potts, Helen E Roy, Christopher Andrews, Catherine M Jones, Jodey M Peyton, Joanna Savage, Martin C Harvey, et al. Monitoring insect pollinators and flower visitation: The effectiveness and feasibility of different survey methods. *Methods in Ecology and Evolution*, 10(12):2129–2140, 2019. 2

[29] Clint RV Otto, Cali L Roth, Benjamin L Carlson, and Matthew D Smart. Land-use change reduces habitat suitability for supporting managed honey bee colonies in the northern great plains. *Proceedings of the National Academy of Sciences*, 113(37):10430–10435, 2016. 1

[30] Luca Pegoraro, Oriane Hidalgo, Ilia J Leitch, Jaume Pellicer, and Sarah E Barlow. Automated video monitoring of insect pollinators in the field. *Emerging Topics in Life Sciences*, 4(1):87–97, 2020. 2

[31] Frank Pennekamp, Nicolas Schtickzelle, and Owen L Petchey. Bemovi, software for extracting behavior and morphology from videos, illustrated with analyses of microbes. *Ecology and evolution*, 5(13):2584–2595, 2015. 2

[32] Alfonso Pérez-Escudero, Julián Vicente-Page, Robert C Hinz, Sara Arganda, and Gonzalo G De Polavieja. idtracker: tracking individuals in a group by automatic identification of unmarked animals. *Nature methods*, 11(7):743–748, 2014. 2

[33] Simon G Potts, Vera Imperatriz-Fonseca, Hien T Ngo, Marcelo A Aizen, Jacobus C Biesmeijer, Thomas D Breeze, Lynn V Dicks, Lucas A Garibaldi, Rosemary Hill, Josef Settele, et al. Safeguarding pollinators and their values to human well-being. *Nature*, 540(7632):220–229, 2016. 1

[34] Gary D Powney, Claire Carvell, Mike Edwards, Roger KA Morris, Helen E Roy, Ben A Woodcock, and Nick JB Isaac. Widespread losses of pollinating insects in britain. *Nature communications*, 10(1):1–6, 2019. 1

[35] Romina Rader, Ignasi Bartomeus, Lucas A Garibaldi, Michael PD Garratt, Brad G Howlett, Rachael Winfree, Saul A Cunningham, Margaret M Mayfield, Anthony D Arthur, Georg KS Andersson, et al. Non-bee insects are important contributors to global crop pollination. *Proceedings of the National Academy of Sciences*, 113(1):146–151, 2016. 1

[36] Malika Nisal Ratnayake, Adrian G Dyer, and Alan Dorin. Tracking individual honeybees among wildflower clusters with computer vision-facilitated pollinator monitoring. *Plos one*, 16(2):e0239504, 2021. 2, 3, 4, 6, 7

[37] Benjamin Risse, Dimitri Berh, Nils Otto, Christian Klämbt, and Xiaoyi Jiang. Fimtrack: an open source tracking and locomotion analysis software for small animals. *PLoS computational biology*, 13(5):e1005530, 2017. 2

[38] Ivan F Rodriguez, Remi Megret, Edgar Acuna, Jose L Agosto-Rivera, and Tugrul Giray. Recognition of pollen-bearing bees from video using convolutional neural network. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 314–322. IEEE, 2018. 3

[39] Orianne Rollin and Lucas A Garibaldi. Impacts of honeybee density on crop yield: A meta-analysis. *Journal of Applied Ecology*, 56(5):1152–1163, 2019. 1

[40] Boris Sekachev, Nikita Manovich, and Andrey Zhavoronkov. Computer vision annotation tool, 2019. GitHub: https://github.com/opencv/cvat. 5

[41] Daniel J Simons and Christopher F Chabris. Gorillas in our midst: Sustained inattentional blindness for dynamic events. *perception*, 28(9):1059–1074, 1999. 1

[42] Tomyslav Sledevič. The application of convolutional neural network for pollen bearing bee classification. In *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, pages 1–4. IEEE, 2018. 3

[43] A Strachecka, M Gryzińska, Magdalena Krauze, et al. The influence of environmental pollution on the protective proteolytic barrier of the honey bee apis mellifera mellifera. *Polish Journal of Environmental Studies*, 19(4):855–859, 2010. 1

[44] William J Sutherland, David B Roy, and Tatsuya Amano. An agenda for the future of biological recording for ecological monitoring and citizen science. *Biological Journal of the Linnean Society*, 115(3):779–784, 2015. 1

[45] Saara Tuohimetsä, Tarja Hietaranta, Marjatta Uosukainen, Sanna Kukkonen, and Saila Karhu. Fruit development in artificially self-and cross-pollinated strawberries (fragaria× ananassa) and raspberries (rubus idaeus). *Acta Agriculturae Scandinavica, Section B—Soil & Plant Science*, 64(5):408–415, 2014. 7

[46] RJ Van Lier, PA Van der Helm, and ELJ Leeuwenberg. Competing global and local completions in visual occlusion. *Journal of Experimental Psychology: Human Perception and Performance*, 21(3):571, 1995. 4

[47] Tristan Walter and Iain D Couzin. Trex, a fast multi-animal tracking system with markerless identification, and 2d estimation of posture and visual fields. *eLife*, 10:e64000, 2021. 2

[48] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 5

[49] Catrin Westphal, Riccardo Bommarco, Gabriel Carré, Ellen Lamborn, Nicolas Morison, Theodora Petanidou, Simon G Potts, Stuart PM Roberts, Hajnalka Szentgyörgyi, Thomas Tscheulin, et al. Measuring bee diversity in different european habitats and biogeographical regions. *Ecological monographs*, 78(4):653–671, 2008. 1

[50] Osamu Yamanaka and Rito Takeuchi. Umatracker: an intuitive image-based tracking platform. *Journal of Experimental Biology*, 221(16), 2018. 2

[51] Cheng Yang and John Collins. A model for honey bee tracking on 2d video. In *2015 International conference on image and vision computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2015. 5

[52] Shivanthan Yohanandan, Andy Song, Adrian G Dyer, and Dacheng Tao. Saliency preservation in low-resolution grayscale images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, 2018. 5

[53] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006. 2, 3, 4