Computers and Electronics in Agriculture 228 (2025) 109665

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Enhancing pollinator conservation: Monitoring of bees through object recognition

Ajay John Alex [a], Chloe M. Barnes [b], Pedro Machado [a], Isibor Ihianle [a], Gábor Markó [c], Martin Bencsik [d], Jordan J. Bird [a,*]

[a] Department of Computer Science, Nottingham Trent University, Nottingham, NG11 8NS, Nottinghamshire, United Kingdom
[b] Department of Applied AI and Robotics, ACAIRA, Aston University, Birmingham, B4 7ET, United Kingdom
[c] Department of Plant Pathology, Hungarian University of Agriculture and Life Sciences, Budapest, 1118, Hungary
[d] Department of Physics, Nottingham Trent University, Nottingham, NG11 8NS, Nottinghamshire, United Kingdom

ARTICLE INFO

Dataset link: https://www.kaggle.com/datasets/birdy654/bee-detection-in-the-wild, https://github.com/AjayJohnAlex/Bee_Detection

Keywords:
Object recognition
Computer vision
Agriculture
Apiculture

ABSTRACT

In an era of rapid climate change and its adverse effects on food production, technological intervention to monitor pollinator conservation is of paramount importance for environmental monitoring and conservation for global food security. The survival of the human species depends on the conservation of pollinators. This article explores the use of Computer Vision and Object Recognition to autonomously track and report bee behaviour from images. A novel dataset of 9664 images containing bees is extracted from video streams and annotated with bounding boxes. With training, validation and testing sets (6722, 1915, and 997 images, respectively), the results of the COCO-based YOLO model fine-tuning approaches show that YOLOv5 m is the most effective approach in terms of recognition accuracy. However, YOLOv5s was shown to be the most optimal for real-time bee detection with an average processing and inference time of 5.1 ms per video frame at the cost of slightly lower ability. The trained model is then packaged within an explainable AI interface, which converts detection events into timestamped reports and charts, with the aim of facilitating use by non-technical users such as expert stakeholders from the apiculture industry towards informing responsible consumption and production.

1. Introduction

The decline of pollinators, particularly bees, has emerged as a critical concern with adverse effects on global food security. In recent times a loss of 1%–10% biodiversity per decade has also been observed (Kluser et al., 2007).

Various species of bee play a key role in agricultural production, supporting a wide array of crops, including fruits, vegetables, oilseeds and legumes, just to name a few; animal pollination supports 30% of global food production (Khalifa et al., 2021). In the United Kingdom alone, 34% of all pollination is provided by one species, the European honey bee (*Apis mellifera*) (Breeze et al., 2011). Factors such as habitat loss and rapid climate change have led to a worrying decline in bee populations around the world, which poses a direct threat to agricultural sustainability.

During these troubling times, the potential of technological intervention through computational intelligence presents a promising approach to mitigate these problems. The notion of Agriculture 4.0 is largely based on data-centric automation (Ampatzidis et al., 2020; Liu et al., 2023; Fountas et al., 2024; Abbasi et al., 2022), which will lead to improvements in agricultural practices in terms of speed and efficiency through the use of technologies such as the Internet of Things (IoT) and edge-based processing, Big Data, Artificial Intelligence (AI) and Machine Learning, as well as Robotics, among others. Precision agriculture applied to apicultural practices can help us autonomously monitor bee behaviour and, in the future, could provide a noninvasive real-time approach to monitor colonies in the long term. This article introduces a novel application of vision-based computational intelligence algorithms towards bee identification and tracking from video streams, which aim to streamline otherwise labour-intensive manual processes. We show that state-of-the-art recognition algorithms such as those within the YOLO suite of research can provide real-time and highly precise bee tracking and that their findings can be distilled into an accessible format for non-technical stakeholders from industry. As shown in Bhuse et al. (2022) and Isa et al. (2022), data augmentation and hyperparameter optimisation are key considerations when aiming for approaches that are useful in the real world beyond the laboratory; the novel combination and application of these state-of-the-art approaches, along with the release of our dataset for future

* Corresponding author.
E-mail address: jordan.bird@ntu.ac.uk (J.J. Bird).

https://doi.org/10.1016/j.compag.2024.109665
Received 17 April 2024; Received in revised form 3 October 2024; Accepted 12 November 2024
Available online 28 November 2024
0168-1699/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

interdisciplinary research, aims to facilitate academic research on the conservation of pollinator populations.

These aforementioned abilities of object recognition algorithms provide timely benefits to saving time and effort in an autonomous monitoring process and further enabling larger-scale bee behaviour analysis. As argued by Ngo et al. (2019), real-time imaging plays a critical role in monitoring honeybee behaviours to assess colony health (Biesmeijer et al., 2006). This is especially important during a time of great habitat loss, insecticide use, and rapid climate change.

In addition to collecting and annotating a novel large-scale dataset, this study also features scientific novelty in the evaluation of several data augmentation and object detection approaches for bee detection. In addition to the data, all algorithms and models are also released as open-source. Literature review reveals that many object detection experiments taking place within precision agriculture and biodiversity studies focus primarily on model ability, while this work also explores the potential for real-time execution. The results show that YOLOv5m emerges as the most accurate approach overall, while YOLOv5s exhibits the most promising performance for real-time use in industry. This study also highlights preliminary results on the significant reduction of inference time for bee detection when using keyframe selection.

Beyond technical achievements, this work encapsulates a broader vision for the future of sustainable beekeeping and the conservation of bee colonies. In addition to the trained machine learning models released as open source, an explainable AI interface is also produced that infers the predictions made by the model and distils them into meaningful and useful information for key stakeholders. Therefore, the addition of this module aims to bridge the gap between complex computational intelligence algorithms and practical apiculture in the real world. Thus, this approach not only improves the state-of-the-art in pollinator observation with a focus on bees, but also contributes to the global agendas of responsible production, climate action, and the preservation of terrestrial ecosystems. The survival of the human species depends on these natural ecosystems.

The scientific contributions arising from this comprehensive exploration of object recognition for bee tracking are a combination of both data and vision models. We collect, annotate, and release a dataset of 9664 images which is open source and available to the research community. In addition, our proposed approach yields several sets of promising results, including the YOLOv5m model, which achieves an mAP@0.5 of 85.6% at 8.1 ms per frame. We also release an explainable AI interface prototype to make the model's findings accessible to non-technical key stakeholders in order to facilitate real-world application and contribute to broader goals of sustainable agriculture and environmental conservation.

Pollinator detection is a task of *small object detection*, which faces several technical challenges in the state of the art (Mirzaei et al., 2023). These include size, given that there are often fewer pixels available that capture the object that is to be detected. Environmental variability is also a concern, since pollinator detection takes place in situ, where lighting and weather conditions can have a significant impact on data distribution. There also exists a trade-off between processing speed and accuracy, where objects must be accurately detected within a timeframe where such detections remain useful for the end user. Towards overcoming these issues, this article proposes the use of data augmentation for increased synthetic variability beyond that which was captured within the dataset, and benchmarking of several YOLO variants to explore the performance vs. computational resource usage trade-off.

The remainder of this article is as follows. A review of the literature is presented in Section 2, covering the relevant work in apiculture and providing a technical background on the techniques applied in this work. The methodology followed in this work is described in Section 3, before the results are discussed in Section 4. Section 5 then suggests future work arising from the findings of this study and finally concludes this article.

2. Background and related work

This section explores relevant state-of-the-art literature in the field. First, we explore work from an agricultural background before describing the state-of-the-art in object detection techniques, which are relevant to the experiments carried out in this article.

2.1. Agriculture, conservation, and biodiversity

Bees are an important pollinator in the global ecosystem, helping plant reproduction and crop quantity and quality. Pollinator services are critical for food production and security; however, bee populations have recently declined. This decline presents a substantial danger to agricultural productivity and the long-term viability of food production (Aizen et al., 2008), which not only contributes significantly to GDP, but maintains the survival of the human species (Klein et al., 2007).

This decline in pollinators, particularly bees, has become a critical concern with adverse effects on global food security. As described in the introduction, the bee plays a key role in the agricultural production of both food and industrial crops. Animal pollination supports 30% global food production (Khalifa et al., 2021), and, beyond food, also plays a significant role in industrial crops. This includes fibres, biofuels, medicinal, material, and ornamental plants (Komlatskiy and Makarova, 2023). Contributing to the wider discussion of environmental sustainability, pollinators, including bees, play a significant role in the cultivation of biofuel crops (Gardiner et al., 2010; Romero and Quezada-Euán, 2013). That is, by facilitating the pollination of biofuel crops such as canola and sunflowers, bees directly improve seed production and thus the availability of these resources, which are critical in sustainable solutions to energy in the global initiative to move away from fossil fuels.

Key stakeholders and beneficiaries, such as farmers in the apicultural field, can implement pollinator-friendly practices that improve agricultural sustainability. For example, avoidance of harmful insecticides (Kremen et al., 2007; Sánchez-Bayo et al., 2016). Some insecticides can be harmful to bees; symptoms include disruption of navigation, feeding behaviour, and reproduction and immune systems (Johansen et al., 1983; Pashte and Patil, 2018). These negative effects result in a decrease in colony survival rates and contribute to population decline (Sánchez-Bayo et al., 2016). Furthermore, climate change has become another risk to bee populations globally (Potts et al., 2016); availability of flowers, nesting locations, and general natural dynamics are negatively affected by the observed rise in temperatures and altered precipitation patterns. Autonomous monitoring of bees can provide additional information on their response to these environmental changes, with the benefit of not requiring human presence throughout the whole data collection process.

This technique contributes to the wider field of environmental monitoring, which is the systematic analysis of the natural environment (Kumar et al., 2012). This can enable researchers to understand the current state of the environment, forecast future trends over time, and ultimately assess the impact of human activity on natural systems. Although this article focuses on pollinator behaviour with a focus on the bee, environmental monitoring also encapsulates, but is not limited to, air, water, and soil quality, as well as biodiversity and climate analysis. In the context of this article, monitoring refers to in situ analysis of bee habitats and behavioural patterns to better understand their ecological importance and contribute to conservation efforts with autonomous analysis and data collection.

Technological intervention to monitor the decline in biodiversity has been promoted in recent times, especially due to the autonomous nature of artificial intelligence and its ability not only to save time and effort, but also to perform tasks for which key stakeholders and beneficiaries simply do not have time to do manually (Ratnayake et al., 2021a; Stark et al., 2023). To this end, researchers suggest that tracking

pollinator behaviour can inform agricultural practices and potentially protect or even improve current crop yield. This could be, for example, information to help optimise crop placement, ensure pollination coverage, and improve the overall efficiency of agricultural systems.

Bee behaviour is often monitored through various modalities of sensor-based data collection and machine learning, such as in Ramsey et al. (2020), where swarming behaviour was recognised from vibro-acoustic accelerometer data at an accuracy greater than 90%. Related work in the field includes (Magnier et al., 2018), which argues that visual alterations, such as background clipping and approximations, can improve pollinator detection through data preprocessing. In Ngo et al. (2019), researchers suggested the use of techniques that include background subtraction, Kalman filtering, and the Hungarian algorithm to produce a system that can monitor the activity that occurs at the entrance of the beehives. The approach achieved around 93.9% accuracy (±1.1%) compared to manual counting. Similar augmentation techniques were proposed in Ratnayake et al. (2021b), where bees in complex outdoor environments were tracked with 86.6% accuracy using the YOLOv2 object detection model.

Environmental monitoring has benefitted from data augmentation as a method to improve the ability to perform automated recognition. In Bittner et al. (2022), the authors propose that the use of GAN-based synthetic data can help improve the detection of various objects present in forest environments to form a more technologically enhanced environmental monitoring system. Kaur et al. proposed a data augmentation strategy to better recognise honeybee disease (Kaur et al., 2022), which used a GAN-based approach to improve classification accuracy over conventional augmentation methods. In De Nart et al. (2022), researchers showed that traditional image augmentation techniques could improve computer vision-based classification of honeybee subspecies, and Buschbacher et al. (2020) also followed a similar methodology in autonomous recognition of bees.

Real-time detection is important for several reasons, such as for purposes of monitoring and intervention. While latency due to algorithm inference is expected, a higher temporal resolution would provide more detailed insight into behavioural patterns. Related literature suggests several figures for the length of time of a pollination activity; Silva et al. (2013) observed that honeybees spend an average of 44.5 (±51.1) seconds with an open flower, suggesting that video matching framerates (i.e., 30 or 60 FPS) may not necessarily require detection to be considered real-time. Results from Chittka et al. (1997) suggest that variability in these figures is high, with two bees from the same species *Bombus pascurorum* spending 7.3 and 2.2 s handling the flowers, respectively. Statistical analyses in the aforementioned study show that handling time differs significantly between species.

The concept of autonomous object detection is at the core of several of the reviewed related works, as well as the main technology behind the approach proposed by this study. This technique is discussed in the following section.

2.2. Object detection

Several related works to this study make use of Bounding Box Object Detection (BBOD) for pollinator monitoring. BBOD is a computer vision technique that predicts the location of a bounding box, or several bounding boxes, around objects of interest. For example, vision systems within an autonomous vehicle aim to draw boundaries or completely segment entities from an image to understand the scene (Amit et al., 2021), such as the detection of an emergency situation where a pedestrian has been detected on a high-speed road.

In Redmon et al. (2016), a model architecture known as You Only Look Once (YOLO) was proposed, with the aim of implementing real-time BBOD. YOLO is a common deep learning architecture for BBOD that performs the detection task as a single regression problem, directly predicting bounding boxes and class probabilities from full input images in one evaluation. This is unlike many other techniques, where proposals are first generated and then later classified iteratively. YOLO operates a grid-based approach, where predictions are made for each grid cell for bounding box coordinates and two confidence scores (one for the box containing the object and one for the class label of the object itself). More details on the differences between model architectures can be found in Jiang et al. (2022).

Several metrics are important for object detection, which are featured in this work due to their importance in the state-of-the-art. They include Intersection over Union (IoU), which is a metric used to evaluate the accuracy of an object detection algorithm on a particular dataset. IoU is the overlap between the predicted bounding box and the ground-truth derived from the expert annotation:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{1}$$

Several different IoU variants, such as GIoU (Generalised IoU) and DIoU (Distance IoU) metrics, have been proposed in the relevant literature to improve the accuracy of object detection algorithms (Zheng et al., 2020). These metrics consider the size and shape of the predicted and ground-truth bounding boxes, as well as their location and orientation.

Similarly to other machine learning problems, precision and recall are also used as part of the evaluation process. Firstly, precision:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}, \tag{2}$$

which is the accuracy of the positive predictions. In this case, precision is the proportion of correctly detected bees out of all instances detected as bees that were incorrect. A *True Positive* denotes a predicted bounding box that contains a bee, and a *False Positive* denotes a predicted bounding box that does not contain a bee.

Recall is also considered:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}. \tag{3}$$

Recall is a measure of the ability of the model to detect all relevant instances. That is, the proportion of true bee instances that were detected by the model out of all instances present in the dataset. *False Negative* in this sense thus denotes a bee in an image that did not receive a predicted bounding box.

Given the trade-off between precision and recall, the mean Average Precision (mAP) is a useful metric in object detection. mAP@0.5 is the mAP calculated at an IoU threshold of 0.5:

$$\text{mAP@0.5} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i|_{\text{IoU}=0.5}, \tag{4}$$

with regards to this work, this means that a prediction is considered a *True Positive* if IoU ≥ 0.5.

Beyond a single threshold, the mAP@0.5:0.95 metric measures mAP over multiple thresholds. In this case, $\{0.5, 0.55, 0.6, \ldots, 0.95\}$:

$$\text{mAP@0.5:0.95} = \frac{1}{10}\sum_{t=0.5}^{0.95} \text{mAP@}t. \tag{5}$$

The use of multiple thresholds allows for a more rigorous approach, by iterating by increasing levels of strictness for overlap, or IoU.

In addition to the prominent metrics for validation, Region of Interest (RoI) is also an important consideration made by the model. A RoI in the example of bee detection may be a flower patch or even anthers, or the entrance to a hive.

Given automatic recognition of RoIs, further analysis can be performed within these areas, and computational resources can be saved by focusing on the parts of the image most likely to contain a bee. Several prominent works in the related literature show that effective RoI detection can significantly reduce the likelihood of false positives from the background (Xiang et al., 2019; Cores et al., 2020). For example, without RoIs, an object detection algorithm trained on a dataset prominent in images of bees collecting pollen may mistakenly classify all flowers as containing bees. This study uses the *You Only Look Once* (YOLO) approach for BBOD. Generally, the YOLO networks

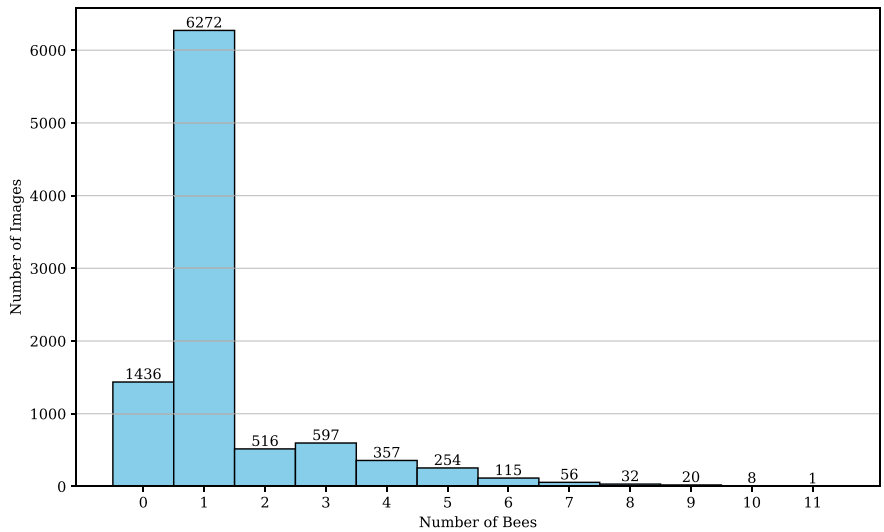
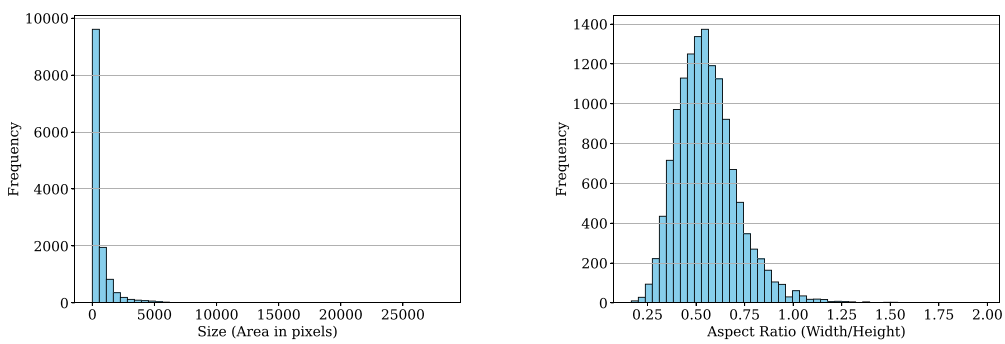


Fig. 1. Presence of the number of bees per image within the dataset.



(a) Distribution of Annotation Sizes (b) Distribution of Aspect Ratios

Fig. 2. Annotation size and bounding box aspect ratio distributions within the dataset.

are generally comprised of three architectures (Jiang et al., 2022; Terven et al., 2023). First, the backbone network which is a Convolutional Neural Network (CNN) trained to extract features from a given image.[1] Features are then further processed at the neck component, a Feature Pyramid Network (FPN) which learns to generate useful feature maps from various levels of the backbone, towards detection of objects at different sizes, and, given the stages of downsampling of the CNN, different levels of abstraction. Finally, the head architecture provides predicted outputs (locations of bounding boxes, classes of objects within those boxes, and confidence scores).

3. Method

This section describes the methods used in this work. This includes data collection and preprocessing (with a link to download the data attributed to this work), augmentation machine learning approaches, and finally a description of the method of farmer-facing inference for accessibility.

3.1. Data collection and preprocessing

Initially, a dataset of images is collected from various locations in the United Kingdom and annotated by several of the authors of this study, with 9664 annotated images that feature a total of 13,402 bounding boxes. On average, there are 1.4 bees per image. Although most of the images (6272) contain one bee, there are a maximum of 11 bees per image. Information derived from the dataset can be found in Table 1. Of the total of 9664 images, 8228 contained at least one instance of a bee. The considerable majority of the images contained one bee, so both the average and standard deviation of bees per image were around 1.39.

Table 1
Information derived from the dataset.

Metric	Value
Total Number of Images	9,664
Total Number of Bees	13,402
Average Number of Bees per Image	1.39
Standard Deviation of Bees per Image	1.39
Maximum Number of Bees in a Single Image	11
Minimum Number of Bees in a Single Image	0
Number of Images with No Bees	1,436

This can be further observed within Fig. 1, which shows the distribution of the number of bees per image within the dataset. Higher counts of bees were less frequent, with only one image containing 11 bees, 8 containing 10 bees, and 20 containing 9 bees.

Fig. 2 shows the distribution of annotation sizes and aspect ratios for the entire dataset. The annotation sizes are relatively consistent for most of the dataset, but it should be noted that there are a number of images that contain larger bounding boxes. Bounding box sizes vary within the dataset, with most annotations covering a standard size

[1] Technical details on the YOLO CNN backbone can be found at: https://github.com/ultralytics/yolov5.

Fig. 3. Examples of annotated (yellow bounding boxes) and preprocessed images within the dataset, selected at random.

and subsets featuring closer and further views, resulting in larger and smaller annotations. The bounding box ratios exhbit variation due to different poses and angles captured within the images.

During the annotation stage, annotators individually checked each image for those that were deemed to be low quality (such as excessive motion blur), removing them from the dataset. The original images are on average 0.92mp (0.18mp to 2.59mp). The original median image ratio is 1280 × 720 px. In the data preprocessing stage, the original data collected for this study were resized to 416 × 416 to meet YOLO input requirements. Bounding box annotations were resized to match via OpenCV's resize function to maintain the spatial relationship after resizing.

Fig. 3 shows examples of annotated images selected from the dataset at random. The dataset contains various bee species including Bombus and Apis mellifera. The dataset generated for this study is available for public download.[2]

3.2. Machine learning and data augmentation

Following data collection and preprocessing, several state-of-the-art object detection algorithms are then benchmarked on the *Bee Detection in the Wild* dataset. This includes YOLOv5, YOLOv5m (variants of frozen and unfrozen weights), YOLOv5s, and YOLOv8m (Ultralytics, 2021).[3]

Given that data augmentation strategies were benchmarked in detail within each of the respective studies for the YOLO-type models, this study thus opts to make use of the default recommended parameters by the original authors. Due to differences in recommended augmentation strategies, pre-processing time was also recorded. Before learning on the full dataset, an initial preliminary exploration is performed into the feasibility of data augmentation; 2000 images are selected at random from the dataset and used to train YOLOv5 both with and without augmentation strategies for 30 epochs to discern whether the strategy could be useful for this problem. A preliminary test is chosen due to the constraints of computational resources, but a more in-depth exploration with the full set could be performed in the future. It is worth noting that augmentation is performed during the training process of the model. That is, the dataset is unchanged in terms of storage, and is augmented in real-time during training.

Following the selection of the data preprocessing strategy, each model is trained for 100 epochs. The aforementioned validation metrics of precision, recall, mAP@.5, mAP@.5:.95, preprocessing time training time, and inference speed are measured and compared. Considerations are given to both model ability in terms of object detection, and also the computational resources required (i.e. comparison of inference time) given that real-world deployment of this model could require recognition frequency to be performed in real-time.

3.3. Stakeholder-facing bee detection and time stamping

An overview of the proposed system can be seen in Fig. 4, which shows an overview of the approach. Furthermore, Fig. 5 shows the prototype system for bee detection and subsequent timestamping to automate the YOLO inference process, implemented with Flask. The interface is web-based and accepts video uploads (such as those collected from an outdoor camera). The system then extracts keyframes from the video at $FPS/2$, infers via the selected model, and finally distils the inference results into a report. The system extracts keyframes at the specified intervals, which are then processed by the object detection model to detect the presence of bees. Detection events are then timestamped, and the number of bees per frame is recorded into a CSV file for visualisation.

4. Results and discussion

This section presents the results for training and validation, as well as the testing of the object detection models on unseen data. Following that, an example of the user interface for accessibility from the stakeholder's perspective is presented, along with pertinent discussions of all the results arising from the experiments carried out in this work.

4.1. Preliminary exploration

An example of data augmentation and its effect on training using a subset of data can be seen in Figs. 6 and 7. Validation losses for

[2] Dataset available from: https://www.kaggle.com/datasets/birdy654/bee-detection-in-the-wild.

[3] Further details on YOLO can be found at: https://github.com/ultralytics/ultralytics.

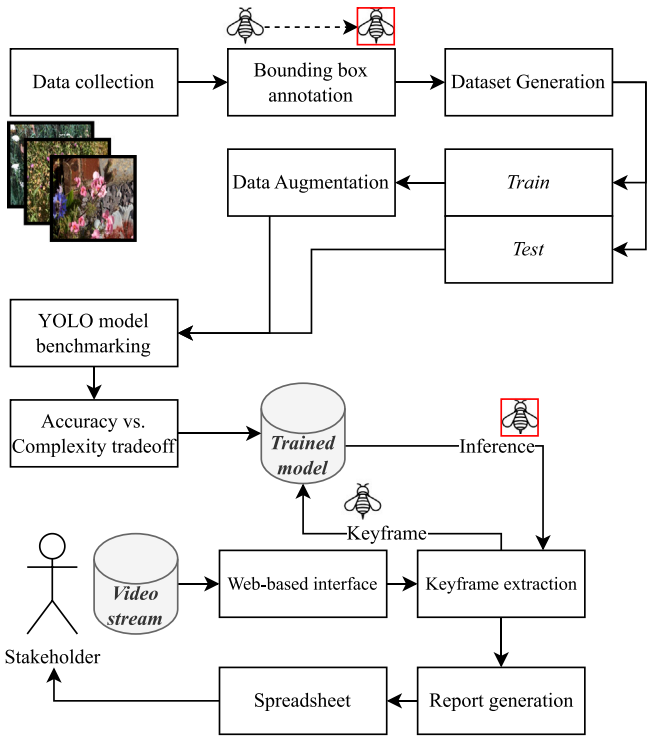


Fig. 4. An overview of the proposed approach for training and inference.

Table 3
Indirect training comparison between YOLOv5 variants with YOLOv8.

Model Summary	YOLOv5s	YOLOv5m	YOLOv8
Precision	81.4	81.0	81.8
Recall	78.6	80.8	80.4
mAP@.5	80.9	81.5	83.3
mAP@.5:.95	34.2	35.6	37.8
Inference Speed (ms)	6.3	7.6	4.2
GPU Training Time (min)	124	163	214

Table 4
Object recognition ability for the models on the testing data. Inference denotes the total average time taken to fully process an input frame.

Model	Precision	Recall	mAP@.5	mAP@.5:.95	Inference (ms)
Base YOLOv5	81.5	76.6	81.4	38.3	4.8
YOLOv5s	82.6	79.7	84.6	41	5.1
YOLOv5m	83.1	81.4	85.6	42.2	8.1
YOLOv5m-frozen	77.2	67.5	75.9	34.7	8.5
YOLOv8m	81.9	80.3	83.3	37.8	12.5

Table 5
Breakdown of average processing time for the models on the testing data.

Model	Average processing time (ms)			
	Pre-process	*Inference*	*NMS*	*Total*
Base YOLOv5	0.1	3.4	1.3	4.8
YOLOv5s	0.1	3.4	1.6	5.1
YOLOv5m	0.1	6.7	1.3	8.1
YOLOv5m-frozen	0.1	6.9	1.5	8.5
YOLOv8m	0.8	10.3	1.4	12.5

Table 2
YOLOv5 model training summary.

Metric	YOLOv5s	YOLOv5m	YOLOv5m-frozen	Base YOLOv5
Precision	81.4	81.0	75.5	80.0
Recall	78.6	80.8	64.8	72.1
mAP@.5	80.9	81.5	69.0	76
mAP@.5:.95	34.2	35.6	27.5	30.6
Inference Speed (ms)	6.3	7.6	8.3	6.9
GPU Training Time (min)	124	163	128	119

both the bounding box (Fig. 6(b)) and objectiveness (Fig. 6(d)) are lower when augmentation strategies are implemented. After 30 epochs using non-augmented and augmented validation data, the bounding-box losses were 0.0415 and 0.0377 respectively (to 3 S.F.). Similarly, for objectiveness loss, non-augmented and augmented validation data losses were 0.00677 and 0.00497 respectively (to 3 S.F.). These results demonstrate the effectiveness of data augmentation during validation.

Interestingly, this effect cannot be seen during training, where the augmented losses are higher for the same metrics (Figs. 6(a) and 6(c) respectively). After 30 epochs, training on nonaugmented and augmented data lead to a bounding-box loss of 0.0122 and 0.0393 respectively, with 0.00282 and 0.00695 reported for objectiveness loss (to 3 S.F.).

4.2. Training results

A summary of the training results of the YOLOv5 series of models can be found in Table 2. It was observed that YOLOv5m has a lower precision than YOLOv5s (81.0 and 81.4, respectively); however, it experiences higher mAP scores for both related metrics (81.5 and 35.6 compared to 80.9 and 34.2, respectively). YOLOv5m with frozen weights has significantly lower recall and mAP values at 64.8, 69.0, and 27.5, respectively. This suggests that the dataset diverges from COCO-related detection activities. Although the Base YOLOv5 model was the quickest to train, the YOLOv5s has a much lower training inference time of 6.3 ms, with a higher speed suggesting that it is more suitable for real-time application.

The results in Table 3 show a comparison between the YOLO models v5 and v8. Since both follow their default, recommended, and different augmentation strategies, the comparison is indirect. Although YOLOv8 has a lower recall score than YOLOv5m, all other metrics outperform it; this includes precision, both mAP scores, and a lower inference speed. It is also worth noting that YOLOv8 takes a considerably longer time to train, suggesting a larger initial investment in computational resources before improving real-time detection ability.

4.3. Testing results

The results in Table 4 show the test results for each of the models. Given these data, it was observed that YOLOv5m has the highest precision at 83.1%, suggesting that it was the best model to identify relevant objects. This approach also scored the highest recall, so it missed the fewest bees when making recognition and classification predictions. At both mAP values, YOLOv5m also scored the highest. Figs. 8 and 9 show examples of mistakes made by the YOLOv5m model. The results show that, even with a precision of 83.1%, the model can miss both relatively small bees (Figs. 8(a) and 8(c)) and relatively large bees (Figs. 8(b) and 8(d)). To this end, examples of misclassified data suggest that additional data could be collected in the future to further improve the robustness of the approach.

Real-time object detection, of paramount importance when transferring research from the laboratory to real-world actionable insights, is the ability to perform inference at suitable speeds as outlined by the problem it is designed to solve; for example, a model that may contribute to a critical decision (such as an autonomous vehicle) should have a low latency time to avoid causing an emergency. In apiculture, a low inference time for bee detection is important for real-time monitoring and responses to changes in activity or health, as well as lower computational complexity making the algorithms more accessible

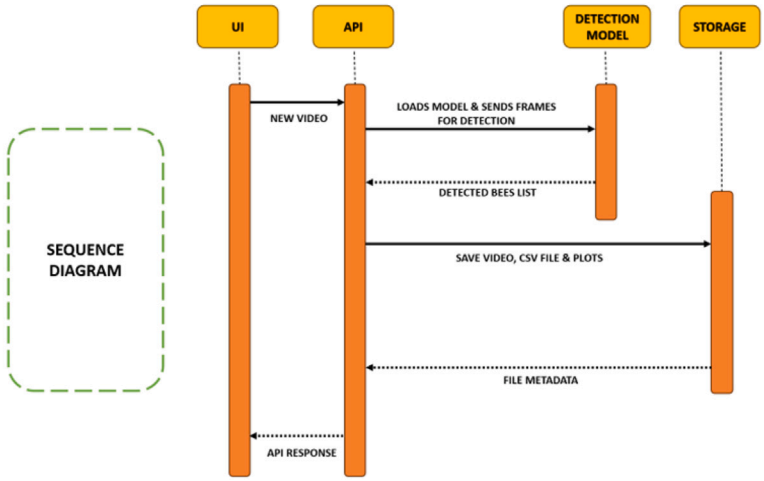


Fig. 5. Graphical overview of the workflow for bee detection and timestamping.

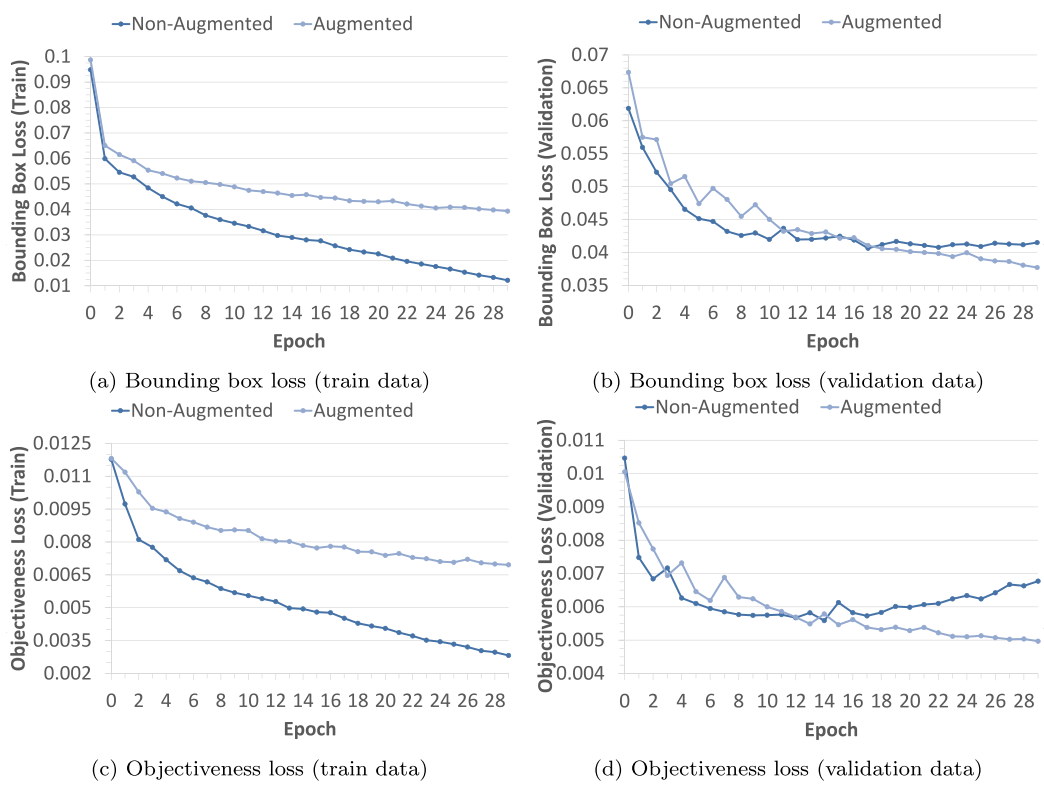


Fig. 6. Metric comparison of bounding box loss (Figs. 6(a) and 6(b)) and objectiveness loss (Figs. 6(c) and 6(d)) for a non-augmented and augmented subset of training and validation data measured over 30 epochs.
Note: y-axis scales are not comparable.

to those who may not have access to high-end computing resources. Table 5 shows a more extensive set of results for time-based metrics. In these experiments, the testing dataset of 996 images was used as input for prediction. The difference in default and recommended augmentation strategies can be seen in the pre-processing time, where YOLOv8 takes 0.7 s compared to 0.3. As also observed, YOLOv8m has the highest inference time, with the lowest being the Yolov5 and YOLOv5s models. Non-Maximum Suppression (NMS) was relatively similar across all models, with the lowest being 1.3 ms (V5 and V5m) and the highest was 1.6 ms (V5s). Overall, the quickest model was Base YOLOv5 at 4.8 ms, closely followed by V5s at 5.1 ms. The highest performing model, YOLOv5m, took longer at 8.1 ms per frame.

4.4. Stakeholder-facing interface

As suggested during the literature review, the use of video streams is showing promise for autonomous monitoring of pollinator behaviour. They are non-intrusive and can be used to capture critical data such as hive health and population decline. Given that Python code and machine learning models are often presented in technical formats, they are therefore inaccessible to a wider audience. Following the training and validation of the models, this article proposes an encapsulation of the work in a format that is accessible to the stakeholder.

Fig. 10 shows the impact of keyframe selection on API response time. As expected, when frames per second are reduced by half, that is,

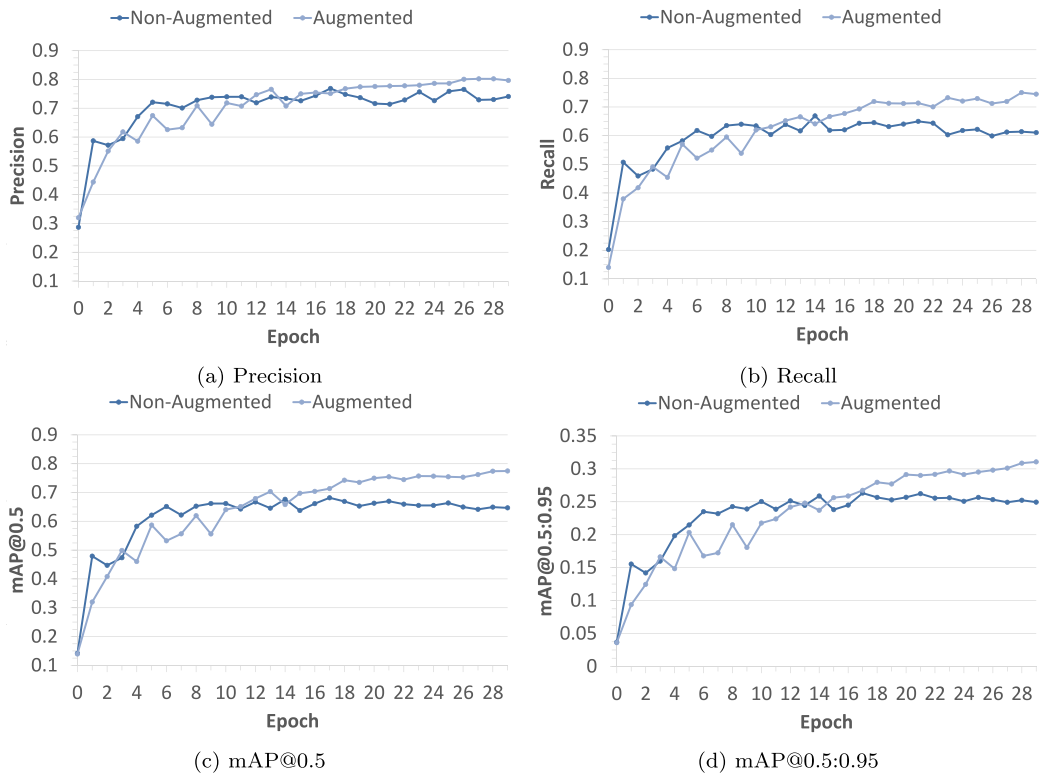


Fig. 7. Metric comparison of (a) precision, (b) recall, (c) mAP@0.5, and (d) mAP@0.5:0.95, for a non-augmented and augmented subset of training and validation data measured over 30 epochs.
Note: y-axis scales for Figs. 7(a) to 7(c) are not comparable to Fig. 7(d).

Table 6
An excerpt of an example report showing the event of a bee entering the camera view and being detected by the object detection model.

D	H	M	S	Video Time	Time (formatted)	DD_HH_MM_SS	Detected
0	0	15	59	959	0 days 0 h 15 mins 59 secs	00:00:15:59	0
0	0	16	0	960	0 days 0 h 16 mins 0 secs	00:00:16:00	0
0	0	16	1	961	0 days 0 h 16 mins 1 secs	00:00:16:01	0
0	0	16	2	962	0 days 0 h 16 mins 2 secs	00:00:16:02	0
0	0	16	3	963	0 days 0 h 16 mins 3 secs	00:00:16:03	0
0	0	16	4	964	0 days 0 h 16 mins 4 secs	00:00:16:04	0
0	0	16	5	965	0 days 0 h 16 mins 5 secs	00:00:16:05	0
0	0	16	6	966	0 days 0 h 16 mins 6 secs	00:00:16:06	1
0	0	16	7	967	0 days 0 h 16 mins 7 secs	00:00:16:07	1
0	0	16	8	968	0 days 0 h 16 mins 8 secs	00:00:16:08	1
0	0	16	9	969	0 days 0 h 16 mins 9 secs	00:00:16:09	1
0	0	16	10	970	0 days 0 h 16 mins 10 secs	00:00:16:10	1
0	0	16	11	971	0 days 0 h 16 mins 11 secs	00:00:16:11	1

$FPS/2$ keyframes from video streams, the processing time is reduced. Videos of 5, 8, 13, and 20 s could be processed in 9, 19, 26, and 38 s, respectively. In the future, keyframe selection strategies could be benchmarked according to the relevant literature, which is discussed in Section 5.

An example of the user interface can be found in Fig. 11. The system encapsulates all model inference and replaces the need to either write Python code or use the command-line interface by enabling the upload of a video file. This video is then automatically processed by the object detection model, and a report is generated along with images of the detected bees for viewing and further analysis. For demonstration purposes, random bee facts are printed onto the screen, but this can be replaced or removed entirely in the future.

Following the submission and processing of a video stream, an example of the aforementioned report can be found in Table 6 with frames selected once per second as an example. These rows are exemplars extracted from the generated CSV file, and show how at 16:06, the model predicted that a bee had entered the frame and thus received a bounding box. It is important to note that this report was generated solely by following three steps: (1) clicking the "Choose File" button, (2) choosing a video file, and (3) clicking the "Detect bees" button. No interaction with code or the command line interface is required, with the aim of democratising the technology arising from the scientific contributions of this work.

5. Conclusion and future work

This study has contributed a significantly large dataset for the research community and explored the application of object detection algorithms for the detection and tracking of bees. The use of these data and/or these algorithms are part of an endeavour to provide a technological intervention in conservation management and analysis. It is particularly pertinent to explore these interventions for pollinators given that food security and the overall survival of the human species is dependent on their services. The democratisation of research is also supported by the approaches proposed by this work, given the open-source nature of all the data, models, and software produced. The

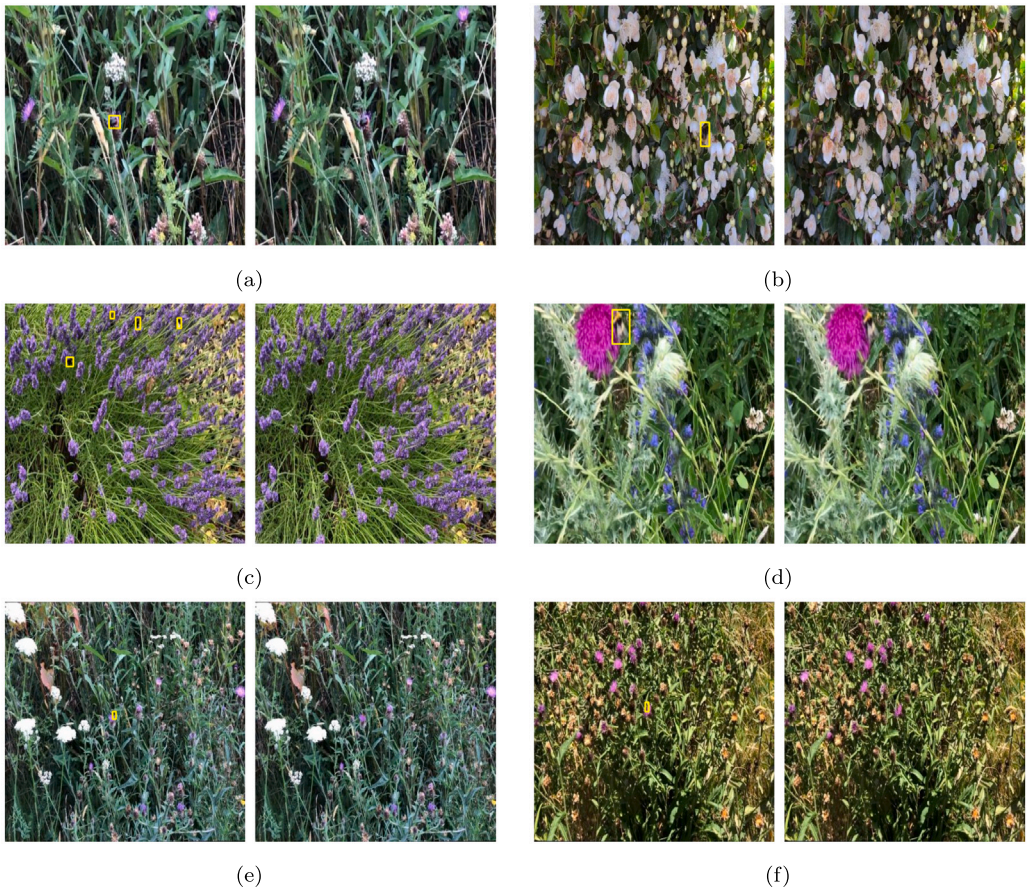


Fig. 8. Examples of ground-truth bees missed by the YOLOv5m model. Left images show ground truth data with yellow bounding boxes, right image shows a lack of model predictions.

findings showed that two particular models benchmarked in this work are promising, the YOLOv5m model which achieved 85.6& mAP@0.5 with an inference time of 8.1 ms per frame, and the more efficient YOLOv5s at 5.1 ms inference at a slightly lower 84.6% mAP. At 5.1 ms per frame, the algorithm is able to execute at approximately 196 frames per second ($\frac{1}{5.1\times10^{-3}\,\mathrm{s}} = \frac{1}{0.0051\,\mathrm{s}} \approx 196.08$).

Beyond data collection and machine learning implementation, this work also argued in favour of stakeholder-friendly and accessible application, and contributed to this in the form of a web application for automated analysis using the selected model. This meant that operators did not need to use any computer code to analyse video streams, such as those collected from an apiary.

A potential scientific limitation of this work lies in the choice of default data augmentation strategies, which were based on the extensive benchmarking performed in the original studies for the YOLO-type models. For additional specificity towards pollinator detection, future work could explore additional data augmentation strategies to better suit this problem. In relation to this, the proposed strategy was restricted to a data subset due to the availability of computational resources. In the future, multiple augmentation strategies could be explored using the full dataset to discern a more general view of which is most effective to use. This, along with combinatorial optimisation of the augmentation parameters, could lead to a better overall model. In terms of real-time processing, a relatively simple keyframe selection strategy of $FPS/2$ was implemented; however, this process could be improved by making use of a more in-depth keyframe selection strategy according to the literature (Rashmi and Nagendraswamy, 2018; Huang and Wang, 2019; Savran Kızıltepe et al., 2023).

In the future, the application of these models in real-world situations, such as an apiary, could provide valuable insight into behavioural analysis. Furthermore, multimodality could improve detection techniques, such as combining audio collected alongside images. Future work could also explore multiple classes, such as the detection of individual species, or behaviours, such as the automatic recognition of pollination events. If used in conjunction with a flower recognition algorithm, this algorithm could then enhance the reporting system towards more detailed information about events that have taken place. Implementation of additional algorithms or types of data would require further work on model architectures beyond those proposed by the original authors.

Towards the next iteration of this work, Fig. 12 shows an example of how the pipeline could be improved. This includes sending notifications to stakeholders and beneficiaries (such as an alert to a population decline) along with the transition away from local to cloud storage. Additionally, the user interface printed random bee facts for demonstration purposes; improvements could be made to the interface in the future, such as replacing this block with a live feed of the detection process, that is, by showing images of some of the bees detected and live graphs that update as the inference process is performed.

Towards integration reflective of Agriculture 4.0, future implementations of this framework could consider wireless communication, such as inference on the edge and wireless communication (5G, 6G,

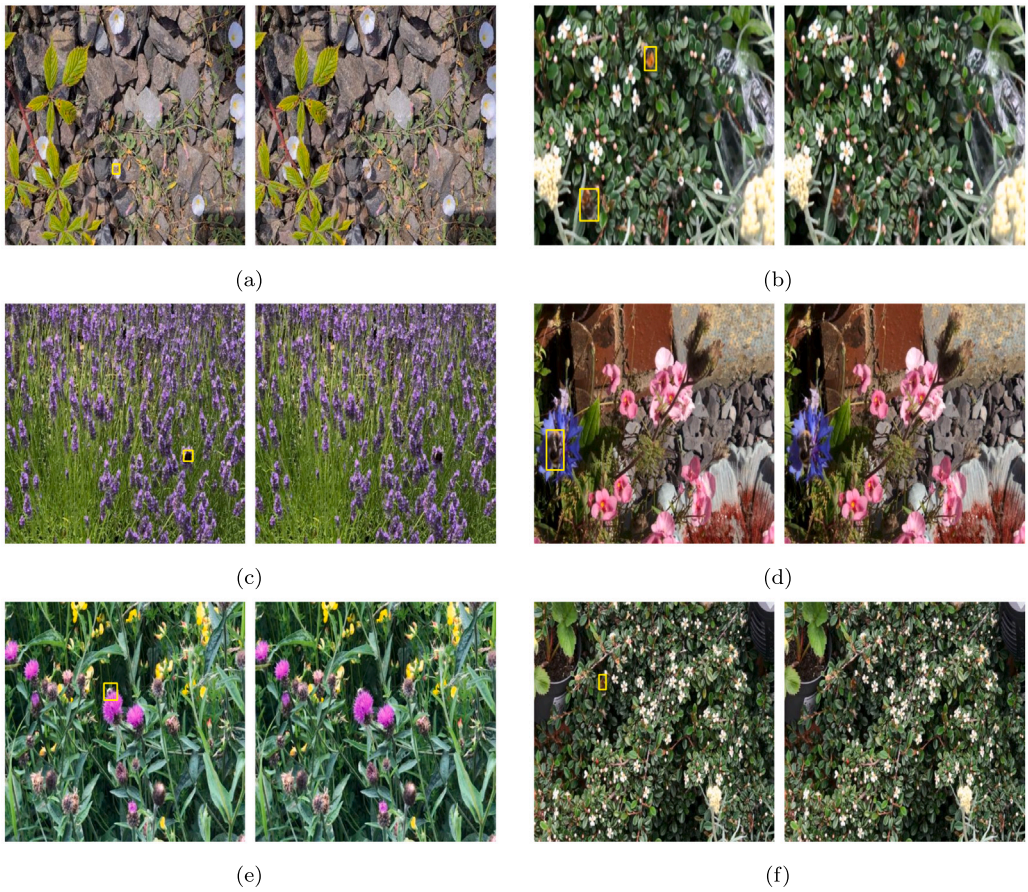


Fig. 9. Additional examples of ground-truth bees missed by the YOLOv5m model. Left images show ground truth data with yellow bounding boxes, right image shows a lack of model predictions.

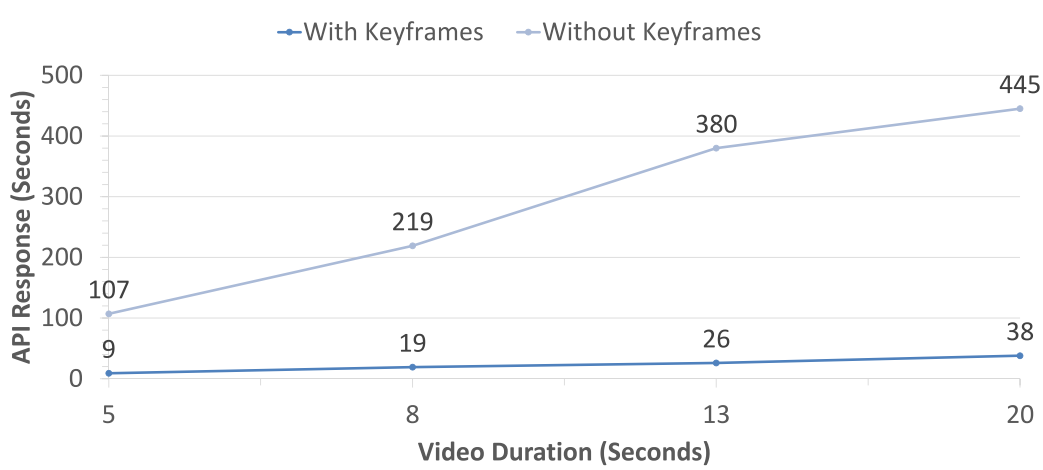


Fig. 10. Impact of keyframe selection on API Response.

LoraWan, etc.). Additionally, enhancements through Geographic Information System (GIS) and Global Positioning System (GPS) data could enable the provision of further context through geographic and location metadata. Furthermore, given long-term data collection and modelling, findings could be used to infer design decisions when building digital twins of the relevant ecosystems.

To finally conclude, this work has proposed the use of a novel dataset and object detection algorithms to facilitate technological intervention in conservation management, with a focus on pollinator

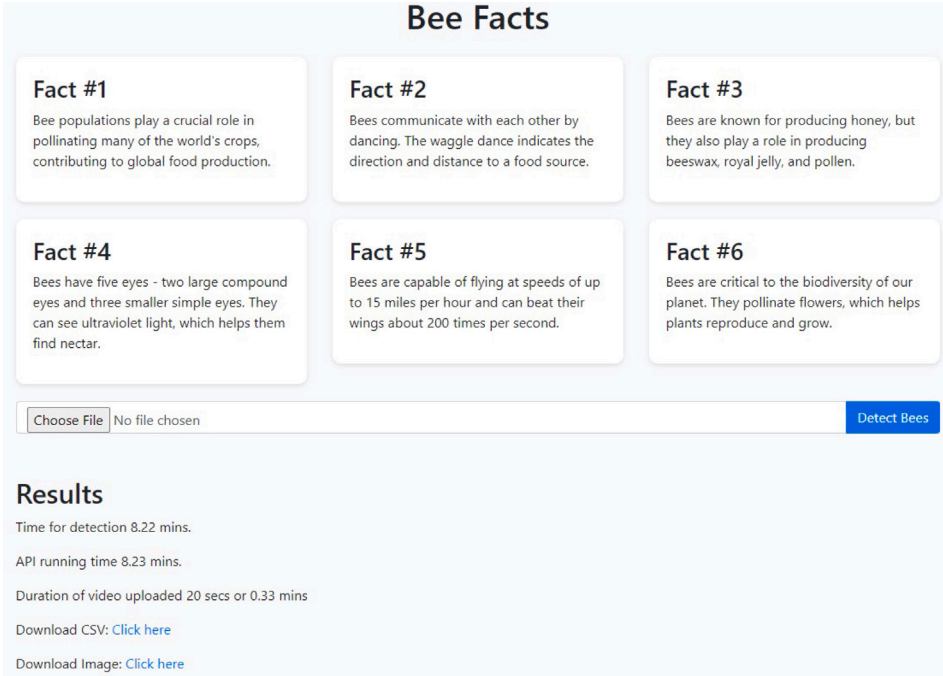


Fig. 11. A screenshot of the interface for the non-technical interaction with the inference model. For demonstration purposes, random bee facts are printed to the screen.

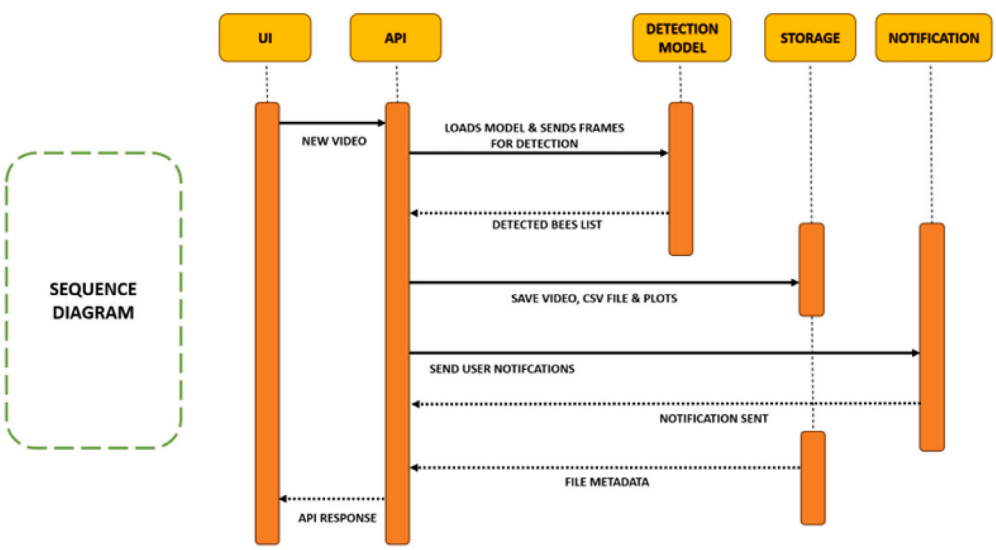


Fig. 12. Proposed future pipeline for improved use in industry.

behaviour. Food security, reacting to climate change, the conservation of the natural world, and responsible consumption and production are critical issues that support human survival, and the use of technology to monitor their improvement is possible given the transfer of interdisciplinary knowledge from the laboratory to the real world.

CRediT authorship contribution statement

Ajay John Alex: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Chloe M. Barnes:** Writing – review & editing, Visualization, Validation, Investigation, Formal analysis, Data curation. **Pedro Machado:** Writing – review & editing, Visualization, Validation, Resources, Methodology, Formal analysis. **Isibor Ihianle:** Writing – review & editing, Visualization, Validation, Resources, Methodology, Formal analysis. **Gábor Markó:** Writing – review & editing, Validation, Investigation. **Martin Bencsik:** Writing – review & editing, Validation, Investigation. **Jordan J. Bird:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data collected and subsequent code written is made publicly available for future work.

The *Bee Detection in the Wild* dataset, collected for and analysed in this study, is released via the Kaggle data science platform under the MIT license. It can be downloaded from: https://www.kaggle.com/datasets/birdy654/bee-detection-in-the-wild.

The code for the web interface used to encapsulate the models is available on Github. It can be downloaded from: https://github.com/AjayJohnAlex/Bee_Detection.

References

Abbasi, R., Martinez, P., Ahmad, R., 2022. The digitization of agricultural industry–a systematic literature review on agriculture 4.0. Smart Agricult. Technol. 2, 100042.

Aizen, M.A., Garibaldi, L.A., Cunningham, S.A., Klein, A.M., 2008. Long-term global trends in crop yield and production reveal no current pollination shortage but increasing pollinator dependency. Curr. Biol. 18 (20), 1572–1575.

Amit, Y., Felzenszwalb, P., Girshick, R., 2021. Object detection. In: Computer Vision: A Reference Guide. Springer, pp. 875–883.

Ampatzidis, Y., Partel, V., Costa, L., 2020. Agroview: Cloud-based application to process, analyze and visualize UAV-collected data for precision agriculture applications utilizing artificial intelligence. Comput. Electron. Agric. 174, 105457.

Bhuse, P., Singh, B., Raut, P., 2022. Effect of data augmentation on the accuracy of convolutional neural networks. In: Information and Communication Technology for Competitive Strategies (ICTCS 2020) ICT: Applications and Social Interfaces. Springer, pp. 337–348.

Biesmeijer, J.C., Roberts, S.P., Reemer, M., Ohlemuller, R., Edwards, M., Peeters, T., Schaffers, A., Potts, S.G., Kleukers, R., Thomas, C., et al., 2006. Parallel declines in pollinators and insect-pollinated plants in Britain and the Netherlands. Science 313 (5785), 351–354.

Bittner, D., Ferreira, J.F., Andrada, M.E., Bird, J.J., Portugal, D., 2022. Generating synthetic multispectral images for semantic segmentation in forestry applications. In: ICRA 2022 Workshop in Innovation in Forestry Robotics: Research and Industry Adoption.

Breeze, T.D., Bailey, A.P., Balcombe, K.G., Potts, S.G., 2011. Pollination services in the UK: How important are Honeybees? Agricult. Ecosyst. Environ. 142 (3–4), 137–143.

Buschbacher, K., Ahrens, D., Espeland, M., Steinhage, V., 2020. Image-based species identification of wild bees using convolutional neural networks. Ecol. Inform. 55, 101017.

Chittka, L., Gumbert, A., Kunze, J., 1997. Foraging dynamics of bumble bees: correlates of movements within and between plant species. Behav. Ecol. 8 (3), 239–249.

Cores, D., Mucientes, M., Brea, V.M., 2020. Roi feature propagation for video object detection. In: ECAI 2020. IOS Press, pp. 2680–2687.

De Nart, D., Costa, C., Di Prisco, G., Carpana, E., 2022. Image recognition using convolutional neural networks for classification of Honey Bee subspecies. Apidologie 53 (1), 5.

Fountas, S., Espejo-García, B., Kasimati, A., Gemtou, M., Panoutsopoulos, H., Anastasiou, E., 2024. Agriculture 5.0: Cutting-edge technologies, trends, and challenges. IT Prof. 26 (1), 40–47.

Gardiner, M.A., Tuell, J.K., Isaacs, R., Gibbs, J., Ascher, J.S., Landis, D.A., 2010. Implications of three biofuel crops for beneficial arthropods in agricultural landscapes. BioEnergy Res. 3, 6–19.

Huang, C., Wang, H., 2019. A novel key-frames selection framework for comprehensive video summarization. IEEE Trans. Circuits Syst. Video Technol. 30 (2), 577–589.

Isa, I.S., Rosli, M.S.A., Yusof, U.K., Maruzuki, M.I.F., Sulaiman, S.N., 2022. Optimizing the hyperparameter tuning of YOLOv5 for underwater detection. IEEE Access 10, 52818–52831.

Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B., 2022. A review of Yolo algorithm developments. Procedia Comput. Sci. 199, 1066–1073.

Johansen, C.A., Mayer, D.F., Eves, J.D., Kious, C.W., 1983. Pesticides and bees. Environ. Entomol. 12 (5), 1513–1518.

Kaur, M., Ardekani, I., Sharifzadeh, H., Varastehpour, S., 2022. A CNN-based identification of Honeybees' infection using augmentation. In: 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering. ICECCME, IEEE, pp. 1–6.

Khalifa, S.A., Elshafiey, E.H., Shetaia, A.A., El-Wahed, A.A.A., Algethami, A.F., Musharraf, S.G., AlAjmi, M.F., Zhao, C., Masry, S.H., Abdel-Daim, M.M., et al., 2021. Overview of bee pollination and its economic value for crop production. Insects 12 (8), 688.

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. Proc. R. Soc. B: Biol. Sci. 274 (1608), 303–313.

Kluser, S., Peduzzi, P., et al., 2007. Global pollinator decline: a literature review. Geneva: UNEP/GRID.

Komlatskiy, G., Makarova, T., 2023. Pollination by bees in industrial crop production. In: BIO Web of Conferences, vol. 66, EDP Sciences, p. 12001.

Kremen, C., Williams, N.M., Aizen, M.A., Gemmill-Herren, B., LeBuhn, G., Minckley, R., Packer, L., Potts, S.G., Roulston, T., Steffan-Dewenter, I., et al., 2007. Pollination and other ecosystem services produced by mobile organisms: a conceptual framework for the effects of land-use change. Ecol. Lett. 10 (4), 299–314.

Kumar, A., Kim, H., Hancke, G.P., 2012. Environmental monitoring systems: A review. IEEE Sens. J. 13 (4), 1329–1339.

Liu, J., Shu, L., Lu, X., Liu, Y., 2023. Survey of intelligent agricultural iot based on 5G. Electronics 12 (10), 2336.

Magnier, B., Ekszterowicz, G., Laurent, J., Rival, M., Pfister, F., 2018. 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. pp. 563–571.

Mirzaei, B., Nezamabadi-Pour, H., Raoof, A., Derakhshani, R., 2023. Small object detection and tracking: a comprehensive review. Sensors 23 (15), 6887.

Ngo, T.N., Wu, K.-C., Yang, E.-C., Lin, T.-T., 2019. A real-time imaging system for multiple Honey Bee tracking and activity monitoring. Comput. Electron. Agric. 163, 104841.

Pashte, V.V., Patil, C.S., 2018. Toxicity and poisoning symptoms of selected insecticides to Honey Bees (Apis mellifera mellifera L.). Arch. Biol. Sci. 70 (1), 5–12.

Potts, S.G., Ngo, H.T., Biesmeijer, J.C., Breeze, T.D., Dicks, L.V., Garibaldi, L.A., Hill, R., Settele, J., Vanbergen, A., 2016. The Assessment Report of the Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services on Pollinators, Pollination and Food Production. Secretariat of the Intergovernmental Science-Policy Platform on Biodiversity ....

Ramsey, M.-T., Bencsik, M., Newton, M.I., Reyes, M., Pioz, M., Crauser, D., Delso, N.S., Le Conte, Y., 2020. The prediction of swarming in Honeybee colonies using vibrational spectra. Sci. Rep. 10 (1), 9798.

Rashmi, B., Nagendraswamy, H., 2018. Effective video shot boundary detection and keyframe selection using soft computing techniques. Int. J. Comput. Vis. Image Process. (IJCVIP) 8 (2), 27–48.

Ratnayake, M.N., Dyer, A.G., Dorin, A., 2021a. Towards computer vision and deep learning facilitated pollination monitoring for agriculture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2921–2930.

Ratnayake, M.N., Dyer, A.G., Dorin, A., 2021b. Tracking individual Honeybees among wildflower clusters with computer vision-facilitated pollinator monitoring. Plos one 16 (2), e0239504.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.

Romero, M.J., Quezada-Euán, J.J.G., 2013. Pollinators in biofuel agricultural systems: the diversity and performance of bees (Hymenoptera: Apoidea) on Jatropha curcas in Mexico. Apidologie 44, 419–429.

Sánchez-Bayo, F., Goulson, D., Pennacchio, F., Nazzi, F., Goka, K., Desneux, N., 2016. Are bee diseases linked to pesticides?—A brief review. Environ. Int. 89, 7–11.

Savran Kızıltepe, R., Gan, J.Q., Escobar, J.J., 2023. A novel keyframe extraction method for video classification using deep neural networks. Neural Comput. Appl. 35 (34), 24513–24524.

Silva, D.P., Moisan-De Serres, J., Souza, D.R., Hilgert-Moreira, S.B., Fernandes, M.Z., Kevan, P.G., Freitas, B.M., 2013. Efficiency in pollen foraging by Honey Bees: Time, motion and pollen depletion on flowers of sisyrinchium palmifolium Linnaeus (Asparagales: Iridaceae). J. Pollinat. Ecol. 11, 27–32.

Stark, T., Ştefan, V., Wurm, M., Spanier, R., Taubenböck, H., Knight, T.M., 2023. YOLO object detection models can locate and classify broad groups of flower-visiting arthropods in images. Sci. Rep. 13 (1), 16364.

Terven, J., Córdova-Esparza, D.-M., Romero-González, J.-A., 2023. A comprehensive review of Yolo architectures in computer vision: From Yolov1 to Yolov8 and Yolo-nas. Mach. Learn. Knowl. Extract. 5 (4), 1680–1716.

Ultralytics, 2021. YOLOv5: A state-of-the-art real-time object detection system. https://docs.ultralytics.com. (Accessed: Feb 2024).

Xiang, C., Yu, Z., Zhu, S., Yu, J., Yang, X., 2019. End-to-end visual grounding via region proposal networks and bilinear pooling. IET Comput. Vis. 13 (2), 131–138.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D., 2020. Distance-IoU loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12993–13000.