

Analyzing Bee Behavior Through Video Tracking Using Computer Vision Techniques

Ian Carlos Rocha Lima¹, André Roberto Ortoncelli¹, Michele Potrich¹, Marlon Marcon^{1*}

Abstract: Bees are responsible for a large part of the pollination of plant species and act as bioindicators in evaluating phytosanitary products. Research on bee behavior and mortality can be carried out in laboratories with a controlled environment. Still, even so, behavioral assessment, when performed visually by an expert, becomes time-consuming and susceptible to failures, which justifies the creation of automated systems for this purpose. This article presents a tool based on Computer Vision to support this activity. The YOLO/Darknet architecture was used to detect bees in videos, and experiments were carried out with eight pre-trained models, obtaining adequate results of insect detection, which were used to develop an interface that provides graphs and reports: heat maps, walking maps, the percentage of the area explored, the time spent moving, and the speed at which individuals move. Experts can interpret the information produced by the proposed tool to understand bee behavior.

Keywords: Insect Detection — Object detection — Heat Maps — Walk Path Analysis

Resumo: As abelhas são responsáveis por grande parte da polinização das espécies vegetais e atuam como bioindicadoras na avaliação de produtos fitossanitários. Pesquisas sobre comportamento e mortalidade de abelhas podem ser realizadas em laboratórios com ambiente controlado. Ainda assim, mesmo assim, a avaliação comportamental, quando realizada visualmente por um especialista, torna-se demorada e suscetível a falhas, o que justifica a criação de sistemas automatizados para essa finalidade. Este artigo apresenta uma ferramenta baseada em Visão Computacional para dar suporte a essa atividade. A arquitetura YOLO/Darknet foi utilizada para detectar abelhas em vídeos, e experimentos foram realizados com oito modelos pré-treinados, obtendo resultados adequados de detecção de insetos, os quais foram utilizados para desenvolver uma interface que fornece gráficos e relatórios: mapas de calor, mapas de caminhada, a porcentagem da área explorada, o tempo gasto em movimento e a velocidade com que os indivíduos se movem. Especialistas podem interpretar as informações produzidas pela ferramenta proposta para entender o comportamento das abelhas.

Palavras-Chave: Detecção de Insetos — Detecção de Objetos — Mapas de Calor — Análise do Trajeto de Caminhada

¹Federal University of Technology – Paraná (UTFPR), Brazil

*Corresponding author: marlonmarcon@utfpr.edu.br

DOI: <http://dx.doi.org/10.22456/2175-2745.143502> • Received: 23/10/2024 • Accepted: 16/01/2024

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

1. Introduction

The order *Hymenoptera* is the third largest species in the class *Insecta*, including ants, wasps, and bees. Because it is home to the largest number of pollinators, it is considered the most important class in terms of conservation of plant and animal species [1]. In Brazil, there are more than 1,500 species of native bees, distributed in almost 300 genera [1] that are responsible for the pollination of 30% to 90% of the native flora, depending on the ecosystem considered [2].

Concerns about bee conservation have intensified recently, as the decline in the population of these pollinators and its economic and ecological consequences have been notable [3].

The abusive use of pesticides and the reduction of the natural habitats of these insects due to deforestation for the planting of mono-cultures have been pointed out as the leading causes of the decrease in bee biodiversity [4]. Studies on these pesticides and the reactions they can cause in target individuals and non-target organisms (such as bees) are essential.

Entomology is a science that studies insects in all aspects, establishing their relationships with humans, plants, and other animals [5]. Within this science, there are divisions, one of which is Agricultural Entomology, responsible for the study of agricultural pests (insects and mites) that cause damage to plants and methods to control them [5]. Within Agricultural Entomology, the objective is for biological control agents

to combat only pests, that is, target organisms. However, there is concern that they may affect non-target organisms, such as beneficial insects, animals, and humans. To this end, researchers develop research with these organisms to evaluate their pathogenic potential (ability to cause disease) in target and non-target organisms.

Among these non-target organisms, we highlight bees. Bees are the main pollinating agents, the remnants of native areas, and the main cultivated plants [1]. These individuals are negatively affected by agricultural control agents. In that case, the entire surrounding ecosystem can be harmed by the lack of pollination, which can decrease the hive's production.

In the case of non-target insects, monitoring their behavior is necessary, traditionally done through visual observation and manual recording by the researcher [6]. Auditing this behavior is essential to identify various information regarding the effects these agents cause on the individual, which demands hours of work and observation, often carried out live by the researcher.

Computer vision methods can be explored to assist researchers who do not need to constantly observe the experiment since software can perform pre-processing and analysis automatically, making it less subjective to errors that the researcher may make.

Given the importance of bees in ecosystems and the significant impact that chemical agents can have on their behavior and survival, it is crucial to develop efficient methods for monitoring and analyzing these effects. Traditional methods, while valuable, are often time-consuming and prone to human error. This article addresses these challenges by proposing a Computer Vision-based tool designed to automate the detection and support behavioral analysis of bees, by leveraging the YOLO/Darknet architecture.

We trained bee detectors with different versions of YOLO, achieving an accuracy of up to 94,14%, which proved sufficient to develop a script capable of adequately identifying different information about bee behavior in a controlled environment: data such as the distance traveled, the percentage of the area explored, the time spent moving, and the speed at which individuals move. In addition, the tool also generates movement maps and heat maps, which indicate the insects' preferred areas of activity. The proposed tool can significantly contribute to understanding the effects of agricultural control agents on bees, helping to preserve these essential pollinators. Our source code is available at: <https://github.com/ICDI/bee-tracking>.

With these objectives in mind, the remainder of this work is organized as follows: Section 2 presents a literature review and background on the topic; Details about the experimental database are in Section 3; The experiments performed with YOLO to detect bees in images are in Section 4; Details about the extracted data and reports produced by our tool are in Section 5; Finally, concluding remarks and future work are in Section 6.

2. Background and Literature Review

The widespread use and availability of machine learning and computer vision techniques allow the development of relatively complex monitoring systems in various domains. Such systems can be used to monitor and inspect the health status of bee colonies, showing potential to identify dangerous conditions before the situation becomes critical and to improve the periodic inspection planning of bee colonies to reduce costs significantly. An in-depth analysis of different applications of computer vision in bee-related problems is in [7]. In this section, we will focus on techniques related to bee tracking.

According to [7], computer vision-based methods for bee classification and detection can be divided into three main groups: i) Conventional computer vision and machine learning techniques, ii) CNN-based classifiers, and iii) CNN-based object detectors. Classical machine learning methods include statistical techniques, feature transformations, support vector classifiers, and simple neural networks.

Deep Convolutional Neural Networks (Deep CNNs) is a particular type of supervised learning. Deep learning methods have shown excellent results, but their main disadvantage is that they require a lot of computation to detect in real-time. However, many computing units for mobile applications have recently emerged, such as the NVIDIA Jetson and accelerators for the Raspberry Pi, which are suitable for online classification. Training, which can be very computationally demanding, can be done externally with specialized hardware [8].

Convolutional neural networks (CNNs) are not only used for classification, but also for object detection in real-world environments. They can automatically identify the main features of an object and differentiate it from the background. This is essential in object detection models, which classify and locate a specific object in an image. The Viola-Jones algorithm [9] was one of the first for object detection, especially with the advancement of deep learning methods in the 2010s; new CNN-based object detectors emerged, such as RCNN [10], SSD [11], and YOLO, which raised the bar for the state-of-the-art. YOLO [12] is one of the most prominent object detection algorithms today, currently in version 10 [13].

We can highlight the work of [14, 15] as methods for detecting insects in a controlled environment. In [14], a technique for tracking multiple fruit flies in a controlled environment (a 24.5 cm platform with a uniform background with a suspended FireWire camera and infrared illumination) is presented, in addition to detecting. The authors also successfully detected multiple bees in a controlled environment. The work of [15, 16] successfully detects various insects, but not unlike our work, they only plot the path of the insects without any additional information, such as a heat map or path history.

Recent work using YOLO to detect bees in different environments can also be highlighted, [17, 16]. The work of [16] detected bees using YOLO and optical flow to monitor the traffic entering and leaving the hives, aiming to provide meaningful information about the hive's health status and the population's size.

In [17], experiments were conducted with different versions of YOLO (YOLOv3, YOLOv5, YOLOv7, and YOLO Representation - YOLOR) to discriminate the three most common alfalfa pollinating bee species, including honeybees, bumblebees, and leafcutter bees, in images collected in a controlled environment. In comparison, YOLOv7 could discriminate bees with an accuracy of 95%.

3. Experimental Database

Twenty-four videos were collected during experiments by the Biological Control group at our university. Each video lasts 10 minutes and involves monitoring a bee placed in a Petri dish. In each video, the bee may have been subjected to a different material depending on the experiment's configuration. All videos were captured with smartphone cameras, approximately 30 cm from the dish. Figure 1 shows frames from videos that make up the experimental dataset.



Figure 1. Examples of frames from videos in the experimental database.

The 24 videos were divided into two groups: 7 videos were used for qualitative validation, while the remaining 17 were employed for quantitative analysis. The frames from these 17 videos were labeled and utilized in the detector's training, validation, and testing phases. Additionally, these 17 videos served as a basis for a data augmentation strategy, creating new videos with 4, 9, and 16 simultaneous bee images. For instance, in the case of a video featuring nine bees, nine distinct 60-second excerpts were selected, with each segment representing a bee in the augmentation process, as illustrated in Figure 2.

In the case of videos with 16 simultaneous bees, 16 60-second segments were selected, but with intersections between them. Each video's different frames were labeled and used to train the detector. It is important to note that the same video

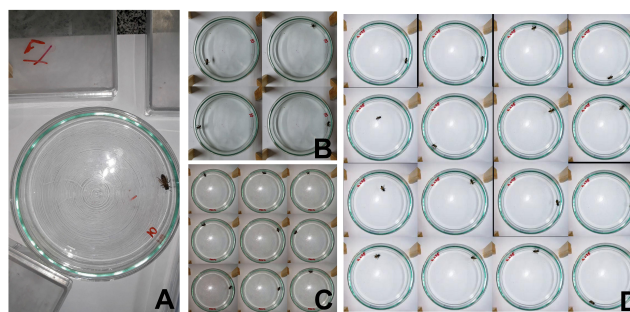


Figure 2. Example of the videos used to construct the dataset: A) shows an original video with only one environment with a bee B) the First type of mosaic, created with four environments; C) the Second type of mosaic, created with nine environments; D) the Third type of mosaic, created with sixteen environments.

is not used in different sets (training, validation, and testing), even when combinations are performed for data augmentation.

The number of videos generated with the data augmentation process, their respective durations, and the number of frames labeled and selected for training, testing and validation are presented in Table 1: the videos column refers to the number of videos used; the bees column, to the number of bees present in each video; the *frames* column refers to the number of frames extracted from all videos; the training, validation and test columns correspond to the number of images used in each subset of the *dataset*.

After selecting the images from the database, they were labeled with the bounding boxes of the objects of interest. The YOLO training process [12] uses these bounding boxes both to learn how to locate the objects contained in them and to evaluate the detection accuracy during the testing phase subsequently.

To carry out the image labeling process, the free web tool ImgLab was employed¹. After uploading the images, each bee was manually delimited with a box/rectangle (dragging the mouse from the top left to the bottom right, as illustrated in Figure 3).

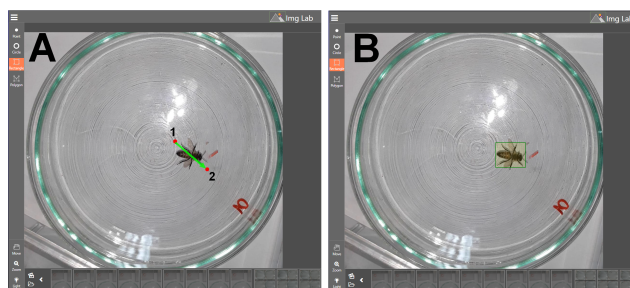


Figure 3. ImgLab tool interface: (A) object marking, with 1 being the starting point of the marking and 2 being the final point of the marking; and (B) final image with the box marked.

¹<https://solothought.com/imglab/>

Table 1. Distribution of images in the database.

Sets	Videos	Bees	Duration	Frames	Training	Validation	Testing
Training	14	1	10:00	480	430	50	0
	4	4	1:30	120	90	30	0
	3	9	1:00	90	60	30	0
	1	16	1:00	30	20	10	0
Testing	3	1	10:00	90	0	0	90
	2	4	1:30	60	0	0	60
	1	9	1:00	30	0	0	30
	1	16	1:00	30	0	0	30
Total	29	-	-	930	600	120	210

For training different versions of YOLO, after marking all the bees in the frames shown in the Table, a file with this data pattern was generated for each image:

- **code**: this is the index of the object in the list of objects to be identified. In the case of this study, the value 0, since it is a single type of target object, the bee;
- **X** and **Y** correspond to the central point of the object on the Cartesian x and y axes, respectively, of the image;
- **width** and **height** correspond to the ratio of the marking size to the image's total size.

4. Bee Detection

During the project's initial phase, image recognition using the *YOLO/Darknet* tool was applied to the training dataset, with different pre-trained architectures, training methods, and image segmentation layers, thus generating a final model with the weights of each architecture used. Different architectures were selected in the official *YOLO/Darknet* repository².

Each of the *CNN*'s architectures was trained with 6 thousand iterations on the training data, which contained 600 images. The following versions of YOLO were trained to detect bees in our database:

- **csdarknet53-omega.conv.105**: is an enhanced version of Darknet-53, which was introduced in YOLOv3. The word refers to an optimization or variation of the cs-darknet53 architecture;
- **yolov4-p5.conv.232**: a variation of the YOLOv4, which uses different settings to adjust the network size (p5). It is a larger version compared to the standard YOLOv4, intended for detection at higher resolutions;
- **yolov4-p6.conv.289**: is a larger variation than p5, generally intended for detection in even larger images or for scenarios where accuracy is prioritized over speed;
- **yolov4.conv.137**: this is the major version of YOLOv4, known for its balance between accuracy and speed. It introduces several improvements, such as Mosaic data augmentation and CIU loss;

- **yolov4-tiny.conv.29**: a scaled-down version of YOLOv4, optimized for resource-constrained devices. It sacrifices some accuracy in favor of higher speed and lower resource usage;
- **yolov7-tiny.conv.87**: a lightweight and fast version of YOLOv7, which is further optimized for low latency and efficient use of resources, while maintaining good performance in object detection tasks;
- **yolov7.conv.132**: one of the latest versions of YOLO, bringing improvements over YOLOv4 and other previous versions. It introduces new optimization techniques that increase both accuracy and efficiency; and
- **yolov7x.conv.147**: a larger and more accurate version of YOLOv7, generally intended for tasks where accuracy is the main focus. It is used for detection in high-resolution images.

Table 2 lists the abbreviation to each YOLO version used in this work, along with their respective number of convolutional layers. The number of layers directly affects the complexity of both training and inference.

To quantitatively evaluate the results obtained with each of the YOLO versions used, the following metrics were used: mean Average Precision (mAP), True Positives (TP), False Negatives (FN), False Positives (FP), and Total of Detections. To compute the map, we use a threshold of 50%, and for the other values, we use 25% with a threshold (YOLO/Darknet default values). All explored YOLO versions were trained with six thousand epochs, and the results obtained are presented in Table 3.

4.1 Analysis of detection results

Based on the mAP values, we selected the four models with the best results to continue the study: *yoloV4-p5*, *darknet-omega*, *yoloV4* and *yoloV7-tiny*.

The model that best adapted to recognize bees was **YOLOv4**, generating a total of 1013 correct detections out of 1080 possible with a total of 1275 detections, having the smallest amount of wasted detections, and having the best results in video detections, one of its competitors, **YOLOv7-tiny**, had a high rate of correct detections, being almost as effective as the

²<https://github.com/AlexeyAB/darknet>

Table 2. YOLO architectures explored in this work

Name	Abbreviation	convolutional layers
csdarknet53-omega.conv.105	darknet-omega	105
yolov4-p5.conv.232	yoloV4-p5	232
yolov4-p6.conv.289	yoloV4-p6	289
yolov4.conv.137	yoloV4	137
yolov4-tiny.conv.29	yoloV4-tiny	29
yolov7-tiny.conv.87	yoloV7-tiny	87
yolov7.conv.132	yoloV7	132
yolov7x.conv.147	yoloV7-X	147

Table 3. Bee detection results

Model	mAP	TP	FP	FN	Total of Detections
darknet-omega	93,92%	992	49	88	1988
yoloV4-p5	93,29%	966	69	114	1798
yoloV4-p6	72,85%	484	18	596	3526
yoloV4	95,03%	1013	36	67	1275
yoloV4-tiny	92,96%	1020	81	60	1463
yoloV7-tiny	93,16%	1018	123	62	2262
yoloV7	90,70%	1011	99	69	1474
yoloV7-X	88,51%	930	99	150	2306

previously mentioned in video detections, even though it was the model with the lowest mAP among those selected, however, it presented a large amount of false positives, 123 in total, generating a total of 1018 correct detections out of 1080 possible with a total of 2262 detections, a little more than double the positives, which generated inconsistencies in the route illustrations and heat map.

The **YOLOv4-p5** and **darknet-omega** models had the lowest TP rates, 966 and 992, respectively; however, when compared to the **YOLOv7-tiny** model, they had approximately half their FP rate, indicating again that there are four effective detection models after full training.

Analyzing the detection results of the trained models and the characteristics of the videos used in the experiment database, we could identify two problems that may have caused the FN and FP: shadow interference and light reflections. Frames with these factors are shown in Figure 4. To obtain even better results in future work, it is essential to properly define a recording protocol that considers lighting conditions and shadow possibilities.

5. Proposed Tool

After training YOLO to detect bees in the images, developing methods to transform the identification and tracking of bees into graphical information began to help researchers extract summarized information from the videos. The first activity is to calculate the center of the region where a bee was detected (Xcenter, Ycenter), the bounding box's centroid. Figure 5 exemplifies a bee's centroid.

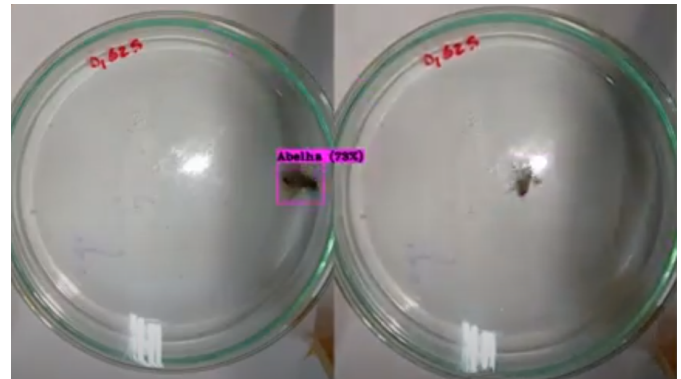


Figure 4. Light interference in detection. In the first image (left), the models could detect the bee. In the second image (right), the bee was close to a light reflection that could not be detected.

After detecting the bee in all video frames, our tool calculates the following metrics: the percentage of the area explored, the time spent moving, and the speed at which individuals move. In addition, our tool also generates movement maps and heat maps.

To generate the heat map highlighting the regions of the image that had the most movement of the bees, we created a matrix the same size as the images with the bees used. This matrix starts with all values equal to zero. For the centroid of each bee identified in each frame of the analyzed video, a value is increased to the matrix at the corresponding position in all points within a radius of 8 pixels of the identified centroid. After performing this process for all frames, the Mat-

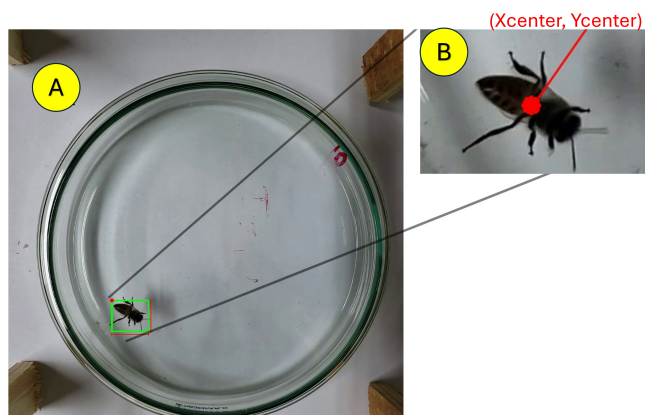


Figure 5. Calculation of the bounding box's centroid: (A) Bounding box of a bee; (B) central point (Xcenter, Ycenter)

matplotlib library³ (from Python) was used, which has graphical resources to generate heat maps from numerical matrices.

Examples of generated heat maps are shown in Figure 6. The four maps shown in Figure 6 refer to the same video but were produced with models trained with different versions of YOLO. The heat maps shown were created with a four-video mosaic, such as the one in part B of Figure 2. In some models, a loss of contrast can be observed, indicating some detection failures.

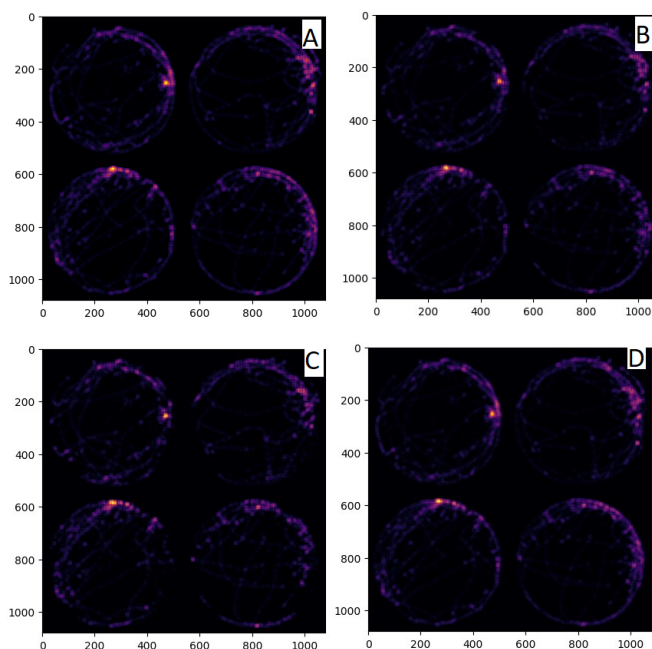


Figure 6. Example of a heat map of bee paths produced with detection results from models trained with different versions of YOLO: (A) YOLOv4; (B) YOLOv7-tiny; (C) YOLOv4-p5; and (D) Darknet-omega

Another graph produced by our tool is related to the move-

³<https://matplotlib.org/>

ment map: the route taken by the insect is presented, using colors that vary from blue to red as temporal information. Pixels with the color closest to blue represent the places where the bee was at the beginning of the video. The pixels closest to red indicate the areas where the bee was at the end of the video. Figure 7 presents a walking map of the same video used in Figure 6.

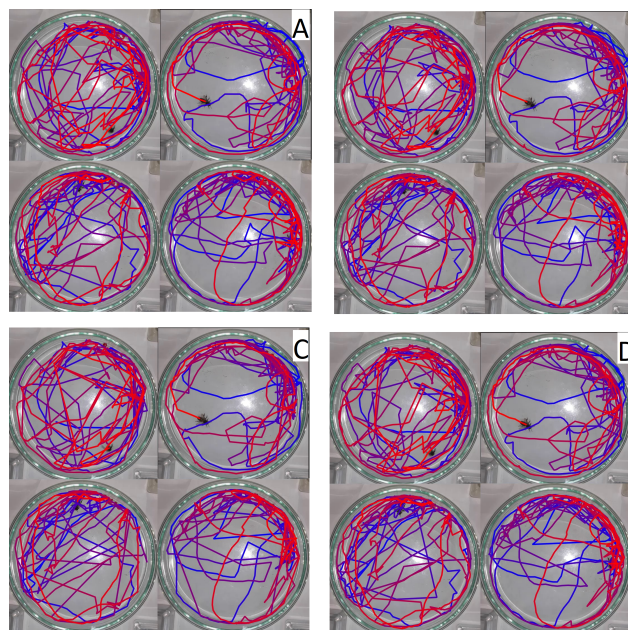


Figure 7. Example of a walking map of bee paths produced with detection results from models trained with different versions of YOLO: (A) YOLOv4; (B) YOLOv7-tiny; (C) YOLOv4-p5; and (D) Darknet-omega

6. Conclusion

In this work, we presented a computer vision-based tool for analyzing bee behavior, utilizing the YOLO/Darknet architecture to detect bees in video footage from controlled experiments. The results obtained from training various YOLO versions have shown promising accuracy, with up to 94.14% detection success. This level of accuracy has allowed us to generate meaningful behavioral data, including metrics such as movement paths, heat maps, and distance traveled by bees. Such data can significantly support researchers in evaluating the effects of agricultural control agents on bees, thereby aiding the preservation of these critical pollinators.

The proposed system demonstrates the potential for automating the analysis of bee behavior, reducing the time and effort required by researchers for manual monitoring and minimizing human error. The tool developed and made available in this work will lead to new research related to the analysis of bee behavior. It may be used by other researchers and developed in future work.

A limitation of the study presented is related to the number of bees in each image. The videos provided for the develop-

ment of this work only included footage of isolated bees in Petri dishes, which prevented a more in-depth study, such as that of [18], in which several individuals are detected in the same environment, so it was not possible to analyze interaction activities between bees during detection. In future work, the tool may be expanded to handle videos of multiple bees adequately.

It is important to remember that none of the neural networks used were developed exclusively for the objectives of this project. For future work, it would be interesting to implement a specific neural network, which may better adapt to the problem addressed. In addition, experiments can also be performed with other versions of YOLO to try to improve the detection results further.

It is also important, in future work, to analyze the tool produced in the opinion of specialists/researchers who work with bees to identify possible improvements and new features that can be added to the reports and graphs produced.

Author contributions

Ian Carlos Rocha: Conceptualization, database preparation, software implementation, execution of experiments. André Roberto Ortoncelli: Software testing and writing of the article. Michele Potrich: Conceptualization, preparation of the database, and review of the article. Marlon Marcon: Conceptualization, software implementation, software, execution of experiments, and writing of the article. All authors read and approved the final version of the text.

References

- [1] NOCELLI, R. et al. Riscos de pesticidas sobre as abelhas. *Semana dos Polinizadores*, v. 3, p. 196–212, 2012.
- [2] KERR, W. E. et al. Aspectos pouco mencionados da biodiversidade amazônica. *Parcerias Estratégicas*, v. 6, n. 12, p. 20–41, 2010.
- [3] SOARES, H. M. Avaliação dos efeitos do inseticida imidacloprido para abelhas sem ferrão scaptotrigona postica latreille, 1807 (Hymenoptera, Apidae, Meliponini). Universidade Estadual Paulista (UNESP), 2012.
- [4] LOPES, I. S. et al. Agrotóxicos: A ameaça de extinção das abelhas no Brasil. *Programa Educativo e Social JC na Escola: Ciência Alimentando o Brasil*, p. 95–110, 2018.
- [5] GALLO, D. et al. Entomologia agrícola. v. 10. *Piracicaba: Fealq*, 2002.
- [6] JORGE, L. d. C. et al. Software para avaliação do comportamento de insetos. *Embrapa Instrumentação-Circular Técnica (INFOTEC-E)*, São Carlos, SP: Embrapa Instrumentação Agropecuária, 2005. 7 p., 2005.
- [7] BILIK, S. et al. Machine learning and computer vision techniques in continuous beehive monitoring applications: A survey. *Computers and Electronics in Agriculture*, Elsevier, v. 217, p. 108560, 2024.
- [8] HORAK, K.; SABLATNIG, R. Deep learning concepts and datasets for image recognition: overview 2019. In: *SPIE. International conference on digital image processing*. [S.l.], 2019. v. 11179, p. 484–491.
- [9] VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: *IEEE. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.], 2001. v. 1, p. I–I.
- [10] GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 580–587.
- [11] LIU, W. et al. SSD: Single Shot Multibox Detector. In: *SPRINGER. European Conference Computer Vision*. [S.l.], 2016. p. 21–37.
- [12] REDMON, J. et al. You only look once: Unified, real-time object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*. [s.n.], 2016. p. 779–788. Disponível em: (<https://doi.org/10.48550/arXiv.1506.02640>).
- [13] HUSSAIN, M.; KHANAM, R. In-depth review of yolov11 to yolov10 variants for enhanced photovoltaic defect detection. In: *MDPI. Solar*. [S.l.], 2024. v. 4, n. 3, p. 351–386.
- [14] BRANSON, K. et al. High-throughput ethomics in large groups of drosophila. *Nature methods*, Nature Publishing Group US New York, v. 6, n. 6, p. 451–457, 2009.
- [15] KIMURA, T. et al. Tracking of multiple honey bees on a flat surface. In: *IEEE. International Conference on Emerging Trends in Engineering and Technology*. 2012. p. 36–39. Disponível em: (<https://doi.org/10.1109/ICETET.2012.25>).
- [16] DICKSON, R. T. et al. Bee traffic estimation with yolo and optical flow. In: *IEEE. SoutheastCon 2024*. 2024. p. 928–933. Disponível em: (<https://doi.org/10.1109/SoutheastCon52093.2024.10500030>).
- [17] ZHANG, C.-J. et al. Evaluation of the yolo models for discrimination of the alfalfa pollinating bee species. *Journal of Asia-Pacific Entomology*, Elsevier, v. 27, n. 1, p. 102195, 2024. Disponível em: (<https://doi.org/10.1016/j.aspen.2023.102195>).
- [18] KIMURA, T. et al. Development of a new method to track multiple honey bees with complex behaviors on a flat laboratory arena. *PloS one*, Public Library of Science San Francisco, USA, v. 9, n. 1, p. e84656, 2014. Disponível em: (<https://doi.org/10.1371/journal.pone.0084656>).