Applying Computer Vision Models to Detect in Real Time the Pollen Flow at the Input of Honeybee Hives (*Apis mellifera* L.)

Daniel de Amaral da Silva[1], **Isac Gabriel Abrahão Bomfim**[3],
Antonio Rafael Braga[2], **Danielo G. Gomes**[1]

[1]Programa de Pós-Graduação em Engenharia de Teleinformática,
Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Centro de Tecnologia, Universidade Federal do Ceará,
Fortaleza - CE, CEP 60.455-970.

[2]Sistemas de Informação, Campus Quixadá,
Universidade Federal do Ceará, Quixadá - CE, CEP 63.902-580.

[3]Laboratório de Apicultura, Campus Crateús,
Instituto Federal do Ceará (IFCE), Crateús-CE.

danielamaral@alu.ufc.br, {rafaelbraga,danielo}@ufc.br, isac.bomfim@ifce.edu.br

Abstract. *Pollen flow into the beehives is directly related to the strength and the health of the honeybee (Apis mellifera L.) colonies. Traditional monitoring of beehives is done through manual and invasive inspections, which are time-consuming, stress bees. On the other hand, the recent paradigm of precision beekeeping has allowed for remote and non-invasive monitoring of hives. In this study, six computer vision models were applied to videos recorded at the entrance of honeybee hives to detect incoming pollen flow. The results showed that the YOLOv7 model performed better in the FPS metric, while also achieving a metric of* AP_{75} *superior to 77%. The CenterNet model presented the best metrics for real-time applications, with an excellent predictive performance at low computational cost.*

1. Introduction

Plants and pollinators, especially bees, have an important ecological relationship that dates back over 120 million years [Khalifa et al. 2021]. Pollination is a crucial process for the reproduction of plant species, with almost 90% of known flowering plant species dependent on animal pollinators to make this process easier [Ollerton et al. 2011]. Pollen, which contains the male gamete for plant reproduction, is also a vital source of nutrients for bees [Nicholls and Hempel de Ibarra 2017]. Given this, honey bees (*Apis mellifera* L.) with their super-populous colonies, have a high demand for this floral resource [Seeley 2006].

Honey bee colonies have workers with defined functions according to their age, and the forager worker has as its main assignment the collection of floral resources, such as nectar and pollen, which ensure the nutrition of the entire colony [Seeley 2006]. The search for pollen by foragers is regulated around the levels of pollen storage and the quantity of larvae in the larval stage being produced at a given moment, whose pheromone released by them is one of the main stimuli for the collection of this food resource

[Fewell 2003]. By observing the pollen stock in the hive, as well as its color variation, the beekeeper can understand the plant sources providing this resource, as well as the health of the colony, and thus, develop appropriate interventions and management practices for it, if necessary [Brodschneider et al. 2021].

Despite the fact that human interventions are crucial for correcting problems and maximizing development and production in commercial apiaries, the constraints of time, distance, and stress on colonies do not allow for daily monitoring [Couto and Couto 2006]. Therefore, less invasive techniques developed for monitoring the flow of pollen-collecting forager bees into hives have proven to be an important tool for beekeepers and precision apiculture [Ngo et al. 2021].

Several recent studies have addressed the challenge of detecting bees carrying pollen on their hind legs through the analysis of hive entrance videos. Some of these works used more traditional computer vision approaches, such as background removal and classification based on SVM or NMC classifiers [Kale et al. 2015, Babic et al. 2016]. Other studies used deep learning techniques, specifically object detection models, such as Faster R-CNN and YOLOv5 [Yang and Collins 2019, Albuquerque et al. 2022]. However, all the cited studies have the limitation of deterministic model selection, as well as the use of hive entrance videos with a fixed camera at the top of the hive, which limits their applicability.

The main objective of this work is to introduce an automated and efficient method for real-time monitoring of pollen influx in bee hives. We will evaluate multiple model candidates, prioritizing general performance and low-power computational requirements, to identify the optimal approach for pollen detection. Additionally, we aim to test the hypothesis that hive entrance video frame characteristics, such as color and texture, significantly impact the model results. This analysis will provide insights into the robustness and adaptability of the proposed models across different environmental conditions.

The implications of this research are significant, particularly in the context of IoT applications. A robust and resource-efficient method for pollen monitoring in bee hives can enhance the overall health and productivity of bee colonies while offering valuable insights to beekeepers and researchers.

2. Materials and Methods

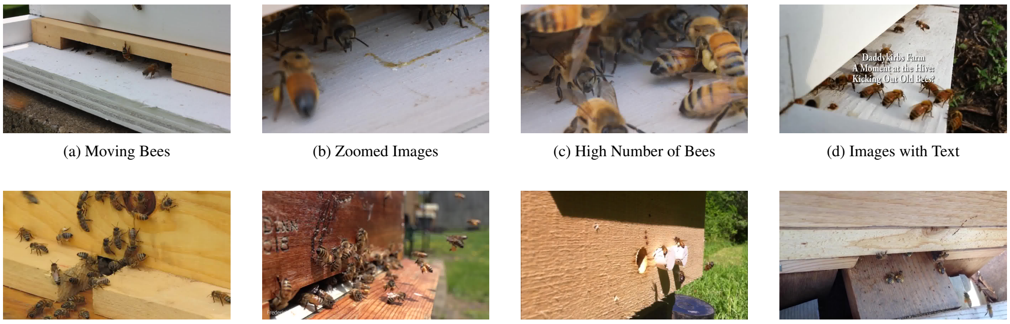
2.1. Image Database

Due to the recent and restricted nature of precision apiculture, well-known and widely disseminated image databases are quite scarce. Numerous hive entrance videos are available on YouTube, however, there are several issues with extracting information, such as low resolution, text in the video, and excessive zoom. This problem ends up reducing the percentage of useful images extracted. Another aggravating factor is the restriction of detecting bees with pollen, a discrete event. Therefore, the experiment depends on several sets of videos for extracting images of bees with pollen.

The images in this work were obtained from a new and unprecedented dataset comprised of 17 selected videos in a YouTube playlist[1]. The playlist was intentionally

[1] https://www.youtube.com/playlist?list=PL_SJHovojluQZ2xRGr3yggi6mTZFiA9UT

created so that the frames extracted from these videos had varied quality and characteristics. In total, about 630,344 images were extracted, of which 1,290 were labeled and constituted the image base used in this experiment. The contrast levels of the image, brightness, and positioning are highly variable in this collection, as shown in Figure 1 (a-h). Our analysis in this article is based on a download of this video list with a resolution of 1280x720, made on December 19, 2022.



(a) Moving Bees (b) Zoomed Images (c) High Number of Bees (d) Images with Text

(e) Similar Color Tones (f) Out-of-Focus Bees (g) Low Image Quality (h) High Image Quality

Figure 1. Different characteristics and variations observed in the input image database of beehives.

Table 1 presents a set of descriptive statistics regarding object size and image characteristics (subsection 2.3) in the dataset. It can be observed that, on average, each image contains more than one object, with an average aspect ratio of 1.17, indicating that the boxes that delimit the bees have a square shape, although some may be wider in width (horizontal axis). In addition, the images present, on average, uniformity in relation to the characteristic variables across the folds. This behavior can be explained by the allocation method of the images in the folds, stratified by video.

Table 1. Descriptive statistics, mean (standard deviation), for object size and image characteristics, overall and stratified by fold.

	Objects				Images		
	Quantity	**Width (px)**	**Length (px)**	**Aspect**	**Color**	**Texture (%)**	**Contrast (%)**
Fold 1	1.49 (0.86)	134.72 (77.31)	119.45 (67.22)	1.16 (0.30)	36.12 (15.90)	32.81 (11.78)	97.73 (3.17)
Fold 2	1.50 (0.87)	138.18 (77.00)	120.54 (70.08)	1.19 (0.29)	35.86 (16.14)	32.85 (12.02)	97.76 (3.31)
Fold 3	1.51 (0.84)	136.50 (83.63)	120.72 (68.37)	1.15 (0.27)	36.26 (16.14)	32.90 (11.89)	97.81 (3.18)
Fold 4	1.40 (0.73)	143.33 (96.21)	123.05 (73.72)	1.17 (0.29)	36.30 (16.02)	33.00 (12.14)	97.66 (3.30)
Fold 5	1.51 (0.86)	140.51 (91.24)	122.01 (72.26)	1.16 (0.28)	36.18 (15.96)	32.81 (12.21)	97.87 (3.08)
Overall	1.48 (0.83)	138.65 (85.29)	121.15 (70.23)	1.17 (0.28)	36.14 (16.00	32.87 (11.98)	97.77 (3.20)

We divided the image dataset into 5 equal partitions for training and validation, known as folds. Each fold was allocated for validation once, allowing for an accurate evaluation of detection model performance. This also allowed us to analyze each fold individually and calculate the mean of the obtained metrics for overall evaluation. This process is known as k-fold cross-validation and was used to obtain an estimate of error throughout the experiment. This approach also allowed for the use of a randomized block design in the statistical analysis of the results.

2.2. Object Detection

Object detection is a complex problem that involves solving two main tasks: classification and regression [Cai and Vasconcelos 2018]. Classification aims to distinguish objects of interest, in our case pollen-laden bees, from the background and assign them the proper class labels. Regression aims to assign precise bounding boxes to different objects. These models can be divided into two basic groups: single-stage models and multi-stage models (two or more) [Zou et al. 2023].

One-stage object detection models are those that perform the task of object detection in a single stage. One-stage object detection models are generally more efficient in terms of time and computational resources, making them potential alternatives for real-time applications [Wang et al. 2023, Zhou et al. 2019]. In this article, we used three members of this group of models: YOLOv7 [Wang et al. 2023], YOLOX [Ge et al. 2021], and CenterNet [Zhou et al. 2019].

The YOLOv7 model presents a specific scaling technique for models that rely on concatenation, resulting in higher efficiency and accuracy compared to earlier models in the YOLO family [Wang et al. 2023]. On the other hand, YOLOX employs the strategy of decoupled head and label processing techniques to improve object detection in overlapping areas [Ge et al. 2021]. Finally, CenterNet uses a pioneering and simple strategy for object detection by predicting central points, making it a model with great potential for real-time applications [Zhou et al. 2019].

The approach of multi-stage object detection models consists of two distinct steps: scanning the image to search for potential regions of the object of interest, and analyzing these regions to determine the presence and identify their exact locations (object cropping) [Cai and Vasconcelos 2018]. These models are known to be more precise and robust than one-stage object detection models, but they are also generally more complex and computationally expensive [Zhu et al. 2020, Zhou et al. 2019]. In this article, we used three members of this group of models: Deformable DETR [Zhu et al. 2020], Faster R-CNN [Ren et al. 2017], and Cascade R-CNN [Cai and Vasconcelos 2018].

Table 2. Settings and optimal hyperparameters of the experimented models obtained via grid search.

Model	Settings*	Learning Rate (α)	Batch Size (β)
YOLOv7	**Backbone:** YOLOv7-X, **Architecture:** P5, **Enhancements:** SyncBN and Mixed Precision	1×10^{-3}	1
YOLOX	**Backbone:** YOLOX-S	1×10^{-3}	2
CenterNet	**Backbone:** ResNet-18, **Enhancements:** DCN	1×10^{-3}	8
Deformable DETR	**Backbone:** R-50, **Enhancements:** Iterative Bounding Box Refinement	1×10^{-4}	2
Faster R-CNN	**Backbone:** R-50, **Enhancements:** FPN	5×10^{-4}	1
Cascade R-CNN	**Backbone:** R-50, **Enhancements:** FPN	1×10^{-3}	2

*: the settings were obtained deterministically, aiming to use the full potential of the available hardware (Subsection 2.5).

Deformable DETR, an extension of DETR, differs from the other models in this experiment by using the Multi-Scale Deformable Attention technique, based on the Transformer architecture [Vaswani et al. 2017], which allows the model to extract features from multiple scales (zoom) simultaneously, enabling more precise detection in terms of details. The second multi-stage model used in the experiment is Faster R-CNN, an extension of Fast R-CNN that introduces the concept of Region Proposal Networks (RPNs). Although the RPN works with different scales and object ratios, Faster R-CNN was used as a baseline model in comparisons. Finally, Cascade R-CNN is a multi-stage extension of Faster R-CNN, where additional detection stages are sequentially concatenated (cascaded) to become more precise.

Table 2 shows the configurations and hyperparameters used for each of the six models studied. Hyperparameter tuning was limited only to the Learning Rate (α), $\alpha \in \{1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 5 \times 10^{-5}, \ldots, 1 \times 10^{-1}; 5 \times 10^{-1}\}$, and Batch Size ($\beta$), $\beta \in \{1, 2, 4, 8\}$, obtained jointly via grid search.

2.3. Metrics

The metrics used in this article can be grouped into two categories: computational performance metrics and predictive performance metrics. In the first category, FPS, FLOPs, and the number of parameters are considered. These metrics are critical to ensuring that a model performs satisfactorily in real-time applications and allow for comparison between different models. In the second category, average precision metrics are considered, which include average precision (AP), average precision with an IoU threshold of 50% (AP_{50}), and average precision with an IoU threshold of 75% (AP_{75}). These metrics are widely used in the analysis of the predictive performance of object detection models.

Finally, to obtain information about the characteristics of the image, we used three metrics: color, texture, and contrast. For the color metric, we used the approach proposed in Hasler and Süsstrunk (2003), the authors define a percentage metric composed of mean and standard deviation operations in the image RGB space. The texture metric was obtained through the absolute gradient of the image, calculated as the mean of the square root of the squared values obtained after applying the Sobel filters ($x + y$). Finally, the contrast metric was calculated using the Michelson algorithm [Moulden et al. 1990].

2.4. Experiment Analysis

For the experiment analysis, we applied a randomized block design, in which the block is given by the fold of the image database, in order to minimize its contribution in comparing the performance of the models. In the general experiments, only the performance of the models was evaluated by blocking for the fold factor. For individual analysis at the fold level, we continued to use a randomized block design, blocking for the fold, but in a factorial experiment with 4 factors: model, contrast, texture, and color.

The normality tests of the residuals were performed using the Shapiro-Wilk test [Shapiro and Wilk 1965], and the O'Neill-Mathews test [O'Neill and Mathews 2000] was used for homogeneity of variances. All the analysis detailed in Section 3 satisfied the assumptions of normality and homogeneity of variances at the 5% significance level.

2.5. Experiment Environment

All experiments involving the training and prediction of the models were carried out on the Google Colab platform in a Python 3.8.1 environment. The models were trained using the mmdetection[2] and mmyolo[3] frameworks with a Pytorch 1.9.0 backend. The statistical analyses were performed using R software version 4.2.1.

[2] https://github.com/open-mmlab/mmdetection

[3] https://github.com/open-mmlab/mmyolo

The server used was specified with an Intel Xeon 2.20GHz CPU. Additionally, the environment had 26 GB of RAM and hosted a Tesla T4 GPU (2560 cores, 16 GB memory).

3. Results

The objective of the study was to evaluate the performance of different object detection models in the task of identifying bees with pollen in images. Six models were tested: Deformable DETR, Faster R-CNN, Cascade R-CNN, YOLOv7, YOLOX, and CenterNet. The overall result of the experiment is shown in Table 3.

Table 3. General results of the six models in the experiment, given by the mean (with standard deviation).

Model	#Params (M)	FLOPs (G)	FPS^{T4}	$\overline{\textbf{AP}}^{val}$	$\overline{\textbf{AP}}^{val}_{50}$	$\overline{\textbf{AP}}^{val}_{75}$
Deformable DETR	40.5	180.0	6.0	**64.7 (2.4)**$^{(a)}$	**95.9 (1.0)**$^{(a)}$	**78.0 (5.3)**$^{(a)}$
Faster R-CNN	41.1	186.9	9.1	56.7 (2.7)$^{(d)}$	92.6 (2.4)$^{(c)}$	65.2 (6.2)$^{(b)}$
Cascade R-CNN	68.9	219.0	6.7	61.4 (2.3)$^{(bc)}$	92.5 (1.6)$^{(c)}$	74.9 (7.6)$^{(a)}$
YOLOv7	71.3	189.9	24.5	63.6 (2.6)$^{(ab)}$	95.1 (1.1)$^{(ab)}$	77.7 (5.6)$^{(a)}$
YOLOX	**8.9**	**30.6**	**41.1**	54.0 (2.5)$^{(e)}$	93.0 (1.2)$^{(bc)}$	57.8 (6.2)$^{(c)}$
CenterNet	14.4	44.5	25.3	59.9 (2.4)$^{(c)}$	93.7 (1.0)$^{(bc)}$	72.9 (4.0)$^{(a)}$

Note 1: The notation $\overline{\textbf{AP}}$ indicates the mean of the **AP** metric evaluated in all folds.
Note 2: Means of the same metric (column) followed by the same letter are statistically equal ($p < 0.05$, Tukey).

Despite the Deformable DETR model achieving the highest average detection precision (AP) at 64.7%, the AP$_{50}$ metric indicates that it can correctly detect 95 bees with pollen out of 100 present in an image, containing at least 50% of the actual bounding box area. Although it has good predictive performance, its high computational cost makes it unsuitable for real-time applications, predicting only 6 frames per second and requiring 180 GFLOPs of computational power.

Similarly, YOLOv7 achieved the second highest average precision (63.6%), a statistically equal value to that found in Deformable DETR, as well as in other predictive performance metrics. However, the big difference and highlight of the model compared to the previous one was its prediction speed, predicting around 24 frames per second, a 300% increase over Deformable DETR, making it a good option for real-time detection.

YOLOv7 did not fully reach weight convergence during training, i.e., at the end of the 50 training epochs, the model still showed a significant drop in the cost function values. Therefore, considering a larger number of epochs, it is possible that YOLOv7 will surpass the metric values found in Deformable DETR.

Although CenterNet did not present the best metrics, it proved to be a balanced model in terms of predictive performance and computational cost, with a maximum difference of 4.8% AP$_{75}$, the most challenging metric in the experiment. Additionally, the model achieved statistically equal performance in the AP$_{50}$ and AP$_{75}$ metrics compared to Deformable DETR and YOLOv7 models. Furthermore, CenterNet had a higher average frame rate (25.3%) using less than 20% of the total parameters compared to YOLOv7. This balanced characteristic makes CenterNet the best overall alternative among the studied models for application in systems with hardware restrictions and requiring considerable predictive performance, as can be seen in its application in Figure 2.

The YOLOX model, statistically presented the worst performance metrics, AP and AP$_{75}$, even being surpassed by the baseline model (Faster R-CNN). Unlike CenterNet, the

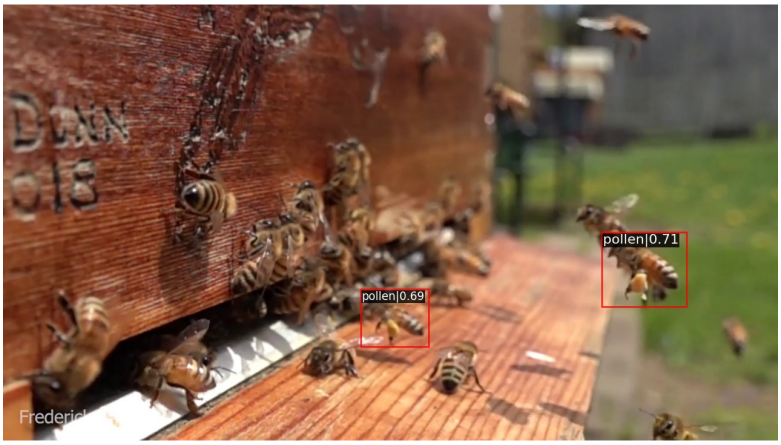


Figure 2. Real-time detection of bees with pollen using the CenterNet model on an image from the test set, with a confidence threshold of 60%.

model only showed good computational cost metrics, surpassing the YOLOv7 model in more than 15 frames predicted per second.

Table 4 shows the results of the metrics of the six models studied in each of the five folds of the image dataset. The results show that the Deformable DETR model presented considerable variations in its performance, especially in the metric of AP_{75}, where it obtained a minimum of 71.7% and a maximum of 85.6%. The model obtained excellent results in the AP metrics in folds 1, 3, and 4, but significantly lost performance in folds 2 and 5, being surpassed by the YOLOv7 and Cascade R-CNN models.

The Cascade R-CNN and YOLOv7 models showed similar metrics in folds 3 and 5, with a maximum variation of less than 1% AP_{75} in fold 5, possibly due to their ability to better handle details in the output. YOLOv7 uses box refinement in the prediction head, while Cascade R-CNN extends Faster R-CNN with additional stages to improve object bounding boxes and reduce the influence of the background. This makes them suitable for applications in images with small details and a predominant background.

Similarly, the CenterNet model achieved satisfactory results in folds 3 and 5, but below expectations compared to Cascade R-CNN, especially in the AP_{75} metric. In these folds, it was expected that CenterNet would perform better, since there is a higher index of objects, thus the probability of object overlap in the image increases.

The Faster R-CNN and YOLOX models did not show significant improvements across folds. Faster R-CNN was used as the baseline model, so superior performance was not expected. YOLOX, despite having metrics below the average of the models, had a maximum difference of only 3.4% compared to YOLOv7 in the conservative metric AP_{50} across folds. This result suggests that YOLOX may be an alternative for applications with extremely restricted hardware configurations (lighter than CenterNet) and that do not require high precision in object cropping.

During the experiments, the analysis of image contrast, color, and texture metrics indicated that these factors did not statistically influence the performance of the models, either individually or collectively. However, it is important to note that the homogeneity of the samples between the folds may have influenced these results, as can be seen in Table 1. Therefore, a deeper investigation at the image level may be necessary to detect more significant differences.

Table 4. Metrics results for the six models in the experiment, stratified by fold.

	#Param. (M)	FLOPs (G)	$\mathbf{AP}^{val}$	$\mathbf{AP}^{val}_{50}$	$\mathbf{AP}^{val}_{75}$
Fold 1					
Deformable DETR	41.0	173.0	**68.4**	**96.4**	**85.6**
Faster R-CNN	41.5	207.1	60.2	93.6	73.3
Cascade R-CNN	44.1	63.3	63.0	92.4	81.7
YOLOv7	71.3	189.9	67.0	95.0	83.5
YOLOX	9.0	26.8	56.6	92.6	64.9
CenterNet	14.1	17.8	63.0	94.9	77.5
Fold 2					
Deformable DETR	41.0	173.0	63.5	94.2	76.4
Faster R-CNN	41.5	207.1	55.8	90.0	65.2
Cascade R-CNN	44.1	63.3	60.8	90.5	76.4
YOLOv7	71.3	189.9	**65.3**	**95.5**	**81.3**
YOLOX	9.0	26.8	54.0	92.0	61.1
CenterNet	14.1	17.8	60.2	93.7	74.8
Fold 3					
Deformable DETR	41.0	173.0	**65.7**	**96.3**	**80.8**
Faster R-CNN	41.5	207.1	55.5	94.0	60.5
Cascade R-CNN	44.1	63.3	61.7	93.1	74.2
YOLOv7	71.3	189.9	60.7	95.1	69.2
YOLOX	9.0	26.8	52.4	93.2	52.5
CenterNet	14.1	17.8	59.8	94.3	74.0
Fold 4					
Deformable DETR	41.0	173.0	**62.7**	**96.5**	71.7
Faster R-CNN	41.5	207.1	53.4	90.1	58.0
Cascade R-CNN	44.1	63.3	57.7	91.7	62.3
YOLOv7	71.3	189.9	61.4	93.4	**75.7**
YOLOX	9.0	26.8	50.9	92.2	50.3
CenterNet	14.1	17.8	56.2	92.1	66.8
Fold 5					
Deformable DETR	41.0	173.0	63.1	95.9	75.6
Faster R-CNN	41.5	207.1	58.4	95.1	69.1
Cascade R-CNN	44.1	63.3	**63.6**	94.7	**79.7**
YOLOv7	71.3	189.9	**63.6**	**96.4**	78.7
YOLOX	9.0	26.8	56.3	95.0	60.3
CenterNet	14.1	17.8	60.1	93.7	71.6

It is important to highlight that this fold analysis was performed with the goal of verifying general differences between the models, but does not exclude the possibility of other image (frame) or video-level analyses to assess the impact of specific characteristics. This result indicates that although feature analysis is important, other metrics and approaches may be necessary for a more comprehensive understanding of model performance.

4. Conclusion

This paper presents an accurate and efficient method for continuously detecting the inflow of pollen in beehives in real time. Analyzing videos of the inflow using state-of-the-art real-time object detection models enabled precise and fast results. In addition to the accuracy and speed in detecting pollen inflow, the YOLOv7 and CenterNet models were found to be robust in different environments, configurations, and camera positions, with no statistically significant differences in results obtained throughout the folds.

The YOLOv7 model is more suitable for environments with less computational cost restrictions, such as cloud environments, while CenterNet can be used locally in low computational cost environments. Thus, the possibility of obtaining precise and fast results, the recommendation of YOLOv7 and CenterNet models as practical tools for real-time detection of pollen inflow in beehives, and the conclusion that visual image characteristics do not affect the efficiency of the method show the robustness and adaptability of the proposed approach for use in real applications.

Building upon this work, we plan to focus on identifying the color patterns of pollen pellets collected by bees at the hive entrance. This will provide valuable insights into the diversity and balance of their diet, as well as the overall health and colony productivity. Another future possibility is to use semi-supervised learning techniques to improve prediction accuracy. This approach is more adaptable in dealing with labeling errors and decreases the need for a large number of training images.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Finance Code 001. Danielo G. Gomes thanks the financial support of the CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) (processes #311845/2022-3 and #432585/2016-8). Antonio Rafael Braga thanks for the financial support of the Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico - FUNCAP, process: BP5-00197-00220.02.00/22.

References

Albuquerque, D., Braga, A., Bomfim, I., and Gomes, D. (2022). Aplicando um modelo yolo para detectar e diferenciar por imagem castas de abelhas melíferas de forma automatizada. In *Anais do XIII Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais*, pages 51–60, Porto Alegre, RS, Brasil. SBC.

Babic, Z., Pilipovic, R., Risojevic, V., and Mirjanic, G. (2016). Pollen bearing honey bee detection in hive entrance video recorded by remote embedded system for pollination monitoring. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-7:51–57.

Brodschneider, R., Kalcher-Sommersguter, E., Kuchling, S., Dietemann, V., Gray, A., Božič, J., Briedis, A., Carreck, N. L., Chlebo, R., Crailsheim, K., et al. (2021). Csi pollen: diversity of honey bee collected pollen studied by citizen scientists. *Insects*, 12(11).

Cai, Z. and Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Couto, R. H. N. and Couto, L. A. (2006). *Apicultura: manejo e produtos*. Funep Jaboticabal.

Fewell, J. H. (2003). Social insect networks. *Science*, 301(5641):1867–1870.

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

Hasler, D. and Suesstrunk, S. E. (2003). Measuring colorfulness in natural images. In Rogowitz, B. E. and Pappas, T. N., editors, *Human Vision and Electronic Imaging VIII*, volume 5007, pages 87–95. International Society for Optics and Photonics, SPIE.

Kale, D. J., Tashakkori, R., and Parry, R. M. (2015). Automated beehive surveillance using computer vision. In *SoutheastCon 2015*, pages 1–3.

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).

Moulden, B., Kingdom, F., and Gatley, L. F. (1990). The standard deviation of luminance as a metric for contrast in random-dot images. *Perception*, 19(1):79–101. PMID: 2336338.

Ngo, T. N., Rustia, D. J. A., Yang, E.-C., and Lin, T.-T. (2021). Automated monitoring and analyses of honey bee pollen foraging behavior using a deep learning-based imaging system. *Computers and Electronics in Agriculture*, 187:106239.

Nicholls, E. and Hempel de Ibarra, N. (2017). Assessment of pollen rewards by foraging bees. *Functional Ecology*, 31(1):76–87.

Ollerton, J., Winfree, R., and Tarrant, S. (2011). How many flowering plants are pollinated by animals? *Oikos*, 120(3):321–326.

O'Neill, M. E. and Mathews, K. (2000). Theory & methods: A weighted least squares approach to levene's test of homogeneity of variance. *Australian & New Zealand Journal of Statistics*, 42(1):81–100.

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149.

Seeley, T. (2006). *Ecologia da Abelha: um estudo de adaptação na vida social (tradução de CA Osowski)*. LTDA, Porto Alegre.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7464–7475.

Yang, C. and Collins, J. (2019). Deep learning for pollen sac detection and measurement on honeybee monitoring video. In *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6.

Zhou, X., Wang, D., and Krähenbühl, P. (2019). Objects as points. *arXiv preprint arXiv:1904.07850*.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276.