AUTOMATIC INSECT COUNTING FOR BIODIVERSITY MONITORING: A REVIEW OF METHODS, CHALLENGES AND FUTURE DIRECTIONS

ORION LIÇI, ANA KTONA

Department of Informatics, Faculty of Natural Sciences,

University of Tirana, Albania

e-mail: orion.lici@fshn.edu.al

Abstract

For better biodiversity conservation we must accurately monitor the insect population. As we already know, insects play a key role as pollinators. This is not, however, the only useful and crucial process in which they are involved. Other useful processes include decomposition, pest control and of course, being a food source for other species (Sánchez-Bayo & Wyckhuys, 2019). It is very disturbing that insect populations are in decline globally. This decline can lead to many environmental consequences, which will subsequently translate into economic problems. Traditional methods of monitoring are labor-intensive, time-consuming, and prone to error. This paper explores the potential of automatic insect counting systems, reviewing the current state of technologies, including visual, optical, acoustic, and hybrid sensor-based methods. Special attention is given to AI-driven approaches such as YOLO (Redmon et al., 2016) and TensorFlow.js, (Smilkov et al., 2019) which enable real-time detection and classification in the field.

Key words: *Biodiversity conservation, AI-Driven methods, YOLO (You Only Look Once); Real-time detection.*

Përmbledhje

Për një ruajtje më të mirë të biodiversitetit, ne duhet të monitorojmë me saktësi popullatën e insekteve. Siç e dimë tashmë, insektet luajnë një rol kyç si pjalmues. Megjithatë, ky nuk është i vetmi proces i dobishëm dhe vendimtar në të cilin ato janë të përfshira. Procese të tjera të dobishme përfshijnë dekompozimin, kontrollin e dëmtuesve dhe sigurisht, të qenit një burim ushqimi për specie të tjera (Sánchez-Bayo & Wyckhuys, 2019). Është shumë shqetësuese që popullatat e insekteve janë në rënie globalisht. Kjo rënie mund të çojë në shumë pasoja mjedisore, të cilat më pas do të përkthehen në probleme ekonomike. Metodat tradicionale të monitorimit kërkojnë shumë *punë, kërkojnë kohë dhe janë të prirura për gabime. Ky punim eksploron potencialin e sistemeve automatike të numërimit të insekteve, duke shqyrtuar gjendjen aktuale të teknologjive, duke përfshirë metodat vizuale, optike, akustike dhe hibride të bazuara në sensorë. Vëmendje e veçantë i kushtohet qasjeve të drejtuara nga IA si YOLO (Redmon et al., 2016) dhe TensorFlow.js, (Smilkov et al., 2019) të cilat mundësojnë zbulimin dhe klasifikimin në kohë reale në terren.*

Fjalë kyçe: *Ruajtja e biodiversitetit, metodat e drejtuara nga IA, YOLO (shikoni vetëm një herë); zbulimi në kohë reale.*

Introduction

Insects can be spotted everywhere. One of their characteristics is the fact that they are the most diverse group of animals on earth. We can count more than a million known insects, and for sure there are many more to be discovered out there (Sánchez-Bayo & Wyckhuys, 2019). They are fundamental to a functional ecosystem. As natural pollinators, they help bio producers to heavily rely on them for better food quality and quantity.

Despite all of these facts, many studies have already proven the sharp decline in insect populations. The factors pertaining to this sharp decline are numerous, among them we can mention loss of habitat, pesticide use and climate change (Sanchez-Bayo & Wyckhuys, 2019). Monitoring their number as well as diversity is fundamental and with this data we can better predict agricultural trends, but also understand ecosystem health. There are also traditional manual methods for monitoring insects that can still be used but there are many limitations.

The main drawback is the requirement for taxonomic expertise, which is very slow and extremely expensive. In addition, they are not feasible for large-scale or long-term monitoring. Automatic insect counting offers a cost-effective solution that can be easily expanded at low cost (Gibb et al., 2019). By combining computer vision, acoustic sensors, and AI we can effectively monitor insects in their environment at reasonable cost. This paper reviews the best practice used for automatic insect counting. and highlights their technological innovations, practical examples and future potential improvements.

1. Methods forautomatic insect counting

1.1 Vision-based methods (computer vision)

Computer vision systems rely on footage and deep learning models to detect and classify insects. For different projects we already have high-resolution cameras installed in the field often as part of camera traps, capturing images or video sequences. Then, we analyze gathered data using object detection algorithms. The most important object detection algorithms we will explore during this paper are YOLO (Redmon et al., 2016; Redmon & Farhadi, 2016; Redmon & Farhadi, 2018; Terven et al. 2023), SSD, and Faster R-CNN (Girshick, 2015; Ren et al., 2016).

YOLO (You Only Look Once) models are highly effective. Thanks to their speed and accuracy, they allow real-time detection in video streams. YOLO models use convolutional neural networks (CNNs) as part of their architecture. Based on this architecture, YOLO offers impressive performance for features like object detection. Its object detection algorithms are focused on shape, color, and pattern detection (Redmon et al., 2016).

Furthermore, real-time detection systems must offer a balance between accuracy and inference speed. In small devices where lack of resources is very crucial, models must be as lightweight and efficient as possible. By using TensorFlow.js, we could run those models in the browser. We can achieve high performance in the browser thanks to WebGL, powering small devices with real-time performance, and avoiding powerful server support (Smilkov et al., 2019).

TensorFlow.js is a JavaScript library that allows for training in and deployment of ML models in the browser and on Node.js. It supports a range of pre-trained models and custom model imports via conversion from Keras or TensorFlow Saved Model formats.

Advantages of computer vision:

- Enables species-level classification.
- Stores visual data for further analysis.
- Allows integration with AI for real-time monitoring (Redmon et al., 2016; Smilkov et al., 2019).

Limitations of computer vision:

- Dependent on lighting and weather conditions.
- Requires extensive labeled datasets for training.
- Struggles with overlapping insects or occlusion (Howard et al., 2017).

1.2 Optical sensor methods

These methods use simple infrared beam systems to detect insect passage. When an insect crosses the beam, it causes a signal interruption, which is recorded. This principle is used in light traps and tunnel-based counting systems (Gibb et al., 2019).

Advantages of optical sensor methods:

- Inexpensive and energy-efficient.
- Suitable for remote and harsh environments.

Limitations of optical sensor methods:

- Cannot identify species or estimate size.
- Prone to false positives from debris or small particles.

1.3 Bioacoustics methods

All insects produce their characteristic noise. From the sound of their wingbeats, to mating calls, or flight noises. Another modern technique is bioacoustics, which identifies and counts insects. It uses microphones and sound analysis techniques combined with AI algorithms. As different species have different sound features, we can train ML models to adapt to only that range of sound features to better target species (Gibb et al., 2019).

Advantages of bioacoustics methods:

- Effective for monitoring night-flying and hidden insects.
- Allow non-invasive observation over large areas.

Limitations of bioacoustics methods:

- Susceptible to ambient noise (wind, rain, animals).
- Difficulty distinguishing overlapping signals.

1.4 Hybrid sensor systems

Each of the outlined methods have their advantages and disadvantages. So, to improve accuracy and reliability we created a system that combines all of these techniques. These hybrid systems combine multiple sensors—vision, acoustics, and environmental. A typical implementation of hybrid combination is AMMOD (Automated Multisensor station for Monitoring Of species Diversity). Prototypes for so-called AMMOD stations are successfully set up in German forests (AMMOD Project, 2022). These are equipped with sensors for recording animal sounds and plant emissions, animal cameras for birds, mammals and insects, and automated insect and pollen sample collectors for monitoring by DNA barcoding. These recordings will be used to generate a solid data pool that will enable the analysis of change and possible trends in species richness and populations. For instance, we want to use the detected species to determine their population and record the change over a long period of time. The visual monitoring subsystems of AMMOD (visAMMOD) consist of automated insect cameras for collecting images of nocturnal moths, which are called moth scanners, and conventional wildlife camera traps for recording observations of insects.

Advantages of hybrid sensor systems:

- Improved detection accuracy.
- Capable of species identification and count.

Limitations of hybrid sensor systems:

- High cost and complexity.
- Requires careful calibration and maintenance (AMMOD Project, 2022).

2. Use of AI and Data analytics

Artificial Intelligence has become a central point to automatic insect counting. Advanced models like YOLOv8, YOLOv12, SSD, Fast R-CNN, and Detectron2 are used to detect insects in images and videos, and offer a high accuracy (Ultralytics, 2024; Liu et al., 2016). These object detection models are also very user friendly. Beyond their object detection they also draw bounding boxes and classify species properly. Training these models requires annotated datasets containing diverse images of insects under different conditions. Transfer learning where pre-trained models are fine-tuned on insect datasets—helps overcome the challenge of limited data.

By using TensorFlow.js we can deploy this model directly in browser. This way we empower not only researchers but also citizens to run AI models on low-powered devices, such as smartphones and laptops without internet access or cloud computing. (Smilkov et al., 2019).

2.1 YOLO (You Only Look Once)

Object detection is now a foundation task in computer vision. It offers machines the feature to identify and localize objects within images or video streams. Early algorithms like R-CNN and its derivatives have an acceptable level of accuracy. However, YOLO (especially later versions) have a higher accuracy rate. Furthermore, YOLO has a lower computational cost due to its innovative architecture. The YOLO family achieved this reduction in computational cost by treating object detection as a single regression problem (Redmon et al., 2016). Also, YOLO evolved through multiple versions, each bringing new features and substantial architecture and performance improvements (Ultralytics, 2024).

2.2 YOLO Direct Export to TensorFlow.js

From 2024, Ultralytics enabled a feature to directly export the model to multiple formats, including TensorFlow.js. This enhancement avoids the need for the extra step needed previously to convert it to ONNX. So, the conversion process became simpler, and the model structure is lightweight and optimized for the browser environment (Ultralytics, 2024).

```
from ultralytics import YOLO
# Load YOLOv8/v11 model
model = YOLO("yolov8s.pt")  # or yolov11s.pt
# Export directly to TensorFlow.js
model.export(format="tfjs")
```

This command generates a TensorFlow.js-compatible model in the following structure:

The folder **'yolov8n_tfjs'** that has two files **'model.json'** and **'group1-shard1of1.bin'**

Then we can load files into the browser using JavaScript:

const myModel = await tf.loadGraphModel('yolov8n_tfjs/model.json');

This feature enables real-time object detection via web interface using a smartphone or webcam.

2.3 SSD (Single Shot MultiBox Detector)

Single Shot MultiBox Detector (SSD), introduced by Liu et al. (2016) and improved for small object detection (Kang & Park, 2023; Shen et al., 2024; Wang, et al., 2024) is another fast object detection framework like YOLO but with notable architectural differences. Like YOLO, SSD performs detection in a single forward pass of the network. However, SSD uses multiple feature maps of different resolutions to detect objects of various sizes, improving the detection of smaller objects—a common limitation in YOLOv1.

SSD defines a set of default bounding boxes (anchor boxes) for each feature map location, with different aspect ratios and scales, allowing the model to predict both object class scores and box refinements simultaneously.

2.4 COCO-SSD

COCO-SSD is another object detection model. Its architecture is based on SSD and as a dataset it uses COCO (Common Objects in Context). The best feature it has that it is lightweight. It is this feature that makes it the perfect choice for browser-based inference using TensorFlow.js, thus making it ideal for real-time applications on client devices without needing powerful server processing (Howard et al., 2017).

Overall, COCO-SSD uses MobileNet as its backbone architecture, which balances accuracy and speed well—ideal for web or mobile apps. It is being used widely in real-time object detection tasks in educational and citizen science applications.

2.5 Comparison between YOLO, SSD, and COCO-SSD

SSD and its COCO-SSD variant represent a parallel lineage to YOLO in the object detection space. However, YOLO, with its latest versions, is dominating the real-time detection market. On the other hand, SSD remains a very competitive alternative, especially for lightweight and browser-based scenarios. Furthermore, COCO-SSD is a ready-to-use model for TensorFlow.js applications, offering a perfect balance between performance and portability. YOLO, SSD and all their variants create a solid tool for a large range of deployment environments. We could use them in cutting edge server environments in addition to web-browser environments (Howard et al., 2017).

Applications Include:

- Real-time insect monitoring in agricultural fields.
- Citizen science and educational tools.
- Rapid biodiversity assessments in remote locations.

Case Study: In a dragonfly monitoring project, YOLOv11 converted to TensorFlow.js enabled species detection and counting through a web interface using the operator's webcam. The inexpensive, browser-based tool allows citizen scientists to contribute to biodiversity monitoring without specialized hardware or software (Ultralytics, 2024).

3. Comparison of methods

Table 1. Methods comparison

Method	Cost	Species ID	Energy Use	Ideal For	Limitations
Computer Vision	Medium–High	Yes	Medium	Monitoring in species-level	Lighting dependency, dataset size
Optical Sensors	Low	No	Low	Abundance tracking in remote areas	No species info, false positives
Bioacoustic Sensors	Medium	Partial	Low	Night insects, hidden spp.	Noise, signal overlap
Hybrid Systems	High	Yes	High	Research, long-term studies	Expensive, complex setup

Each method must be chosen according to the project goal — species-level biodiversity research or high-throughput abundance monitoring.

4. Challenges and future directions

Insect classification is highly complex due to fine-grained visual differences, high similarity of inter-class, and limited datasets (Wäldchen et al., 2018). Insect species often have very slight differences in wing shape, coloration, or size that are very hard to detect in low quality images or with motion blur in real-time scenarios. Solving these problems requires high-quality devices, effective augmentation techniques, and well-trained models.

Future work should be focused on creating a large dragonfly dataset. Train the YOLO with this dataset and export the model to TensorFlow.js platform (Ultralytics, 2024). Furthermore, we could also add geolocation and other environmental metadata to create a full picture of the ecological panorama.

Despite significant progress, challenges remain:

- **Data Availability:** High-quality, annotated insect datasets are scarce.
- **Model Generalization:** AI models trained in one region/species may not perform well in others.
- **Energy Consumption:** Field-deployed systems need low-power solutions.
- **Standardization:** Lack of protocols for comparing methods across studies.
- **Scalability:** Need for scalable infrastructure that integrates sensor data, AI, and IoT.

Future Research Should Focus On:

- Developing large, open-source insect datasets.
- Creating energy-efficient embedded devices.
- Using transfer learning and unsupervised learning.
- Building cloud-connected systems for real-time monitoring.
- Combining audio, visual, and environmental data using multimodal AI (Smilkov et al., 2019; Gibb et al., 2019).

Conclusions

Machine learning models used for insect counting will radically improve ecological research. It covers the gap of limitations created by traditional manual methods and opening new possibilities for real-time, scalable monitoring.

Advanced Machine Learning technologies like YOLO and TensorFlow.js produce lightweight models and, in combination with hybrid multisensory systems, are the best solution (Redmon et al., 2016; Smilkov et al., 2019). For a more economically feasible solution we can avoid hybrid systems and focus more on one of the following technologies:

1) Vision-Based Methods (Computer Vision)

2) Optical Sensor Methods

3) Bioacoustics Methods

Another key factor to success is interdisciplinary collaboration. For a better-tuned system, computer scientists must identify the most notable features of each species in collaboration with ecologists, engineers and data analysts. Based on that feature, the best solution can be implemented.

From these significant improvements in machine learning, IoT, and sensor technologies, new opportunities emerge. With these opportunities, automated insect monitoring will become an essential tool in safeguarding biodiversity.

Real-time image detection for dragonfly species based on TensorFlow.js offers a better approach in ecological monitoring and biodiversity research. By including such web-based technologies, researchers and citizens can access this tool without needing a high-end server infrastructure.

This literature review has highlighted the main object detection models such as SSD, YOLO and COCO-SSD. With the rise of TensorFlow.js and improvements in model compression, the future of browser-based species detection is promising. This will also contribute to insect biodiversity studies. Continuous research and model training will be the key points to improving the detection process in the field.

References

AMMOD Project. (2022). Automated Multisensor station for Monitoring Of species Diversity. https://ammod.de

Gibb, R., Browning, E., Glover-Kapfer, P., & Jones, K. E. (2019). Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring. Methods in Ecology and Evolution, 10(2), 169–185. https://doi.org/10.1111/2041-210X.13101

Girshick, R. (2015). "Fast R-CNN", IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440-1448, doi: 0.1109/ICCV.2015.169.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreeto, H., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

Kang, S.-H. & Park, J.-S. (2023). Aligned Matching: Improving Small Object Detection in SSD. Sensors, 23, 2589. https://doi.org/10.3390/s23052589

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In Computer Vision – ECCV 2016 (pp. 21–37). Springer. https://doi.org/10.1007/978-3-319-46448-0_2

Redmon, J. & Farhadi, A. (2016). YOLO9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242. https://arxiv.org/abs/1612.08242

Redmon, J. & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767. https://arxiv.org/abs/1804.02767

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788. https://doi.org/10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497. https://arxiv.org/abs/1506.01497

Sanchez-Bayo, F. & Wyckhuys, K. A. G. (2019). Worldwide decline of the entomofauna: A review of its drivers. Biological Conservation, 232, 8–27.

https://doi.org/10.1016/j.biocon.2019.01.020

Shen, Y., Xue, J., & Song, A. (2024). Small-object detection algorithm based on improved SSD (SSDRes50-AF). In Proceedings of SPIE (Vol. 13181). SPIE.

https://doi.org/10.1117/12.3031256

Smilkov, D., Thorat, N., Nicholson, C., Reif, E., Viégas, F., & Wattenberg, M. (2019). TensorFlow.js: Machine Learning for the Web and Beyond. arXiv preprint arXiv:1901.05350. https://doi.org/10.48550/arXiv.1901.05350

TensorFlow.js documentation. https://www.tensorflow.org/js

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. Machine Learning and Knowledge Extraction, 5(4), 1680–1716.

https://doi.org/10.3390/make5040083

Ultralytics. (2024). YOLO Docs. https://docs.ultralytics.com

Wäldchen, J., Mäder, P., & Rzanny, M. (2018). Automated plant species identification—Trends and future directions. PLOS Computational Biology, 14(4), e1005993.

https://doi.org/10.1371/journal.pcbi.1005993

Wang, L., Shoulin, Y., Alyami, H., Laghari, A.A., Rashid, M., Almotiri, J., & Alturise, F. (2024). A novel deep learning-based single shot multibox detector model for object detection in optical remote sensing images. Geoscience Data Journal, 11, 237–251. Available from: https://doi.org/10.1002/gdj3.162