

# IBBI: A Python package for the detection and classification of bark and ambrosia beetles

Christopher Marais <sup>1</sup>, Eric Kuo  <sup>1</sup>, Jiri Hulcr  <sup>1</sup>, and Raquel Dias  <sup>2</sup>

<sup>1</sup> School of Forest, Fisheries, and Geomatics Sciences, University of Florida, Gainesville, FL, USA  
<sup>2</sup> Microbiology and Cell Science, University of Florida, Gainesville, FL, USA

DOI: [10.21105/joss.08796](https://doi.org/10.21105/joss.08796)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

---

Editor: [Nick Golding](#)  

Reviewers:

- [@PetervanLunteren](#)
- [@martibosch](#)

Submitted: 26 June 2025

Published: 21 December 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

The Intelligent Bark Beetle Identifier (IBBI) is an open-source Python package that provides a simple, unified interface for the automated detection and classification of bark and ambrosia beetles (Coleoptera: Curculionidae: Scolytinae and Platypodinae) from images. This package addresses a critical need in forest health and biodiversity research by leveraging multiple state-of-the-art computer vision models to streamline a traditionally labor-intensive and expert-dependent task. The target audience for `ibbi` includes entomologists, ecologists, forest health professionals, and biodiversity researchers who require high-throughput identification of beetle specimens from field or lab images. By providing programmatic access to a suite of pre-trained models for object-detection and species classification, `ibbi` empowers researchers to accelerate data analysis for ecological studies, invasive species monitoring, and biodiversity assessments. The core functionality is accessible through a single factory function, `create_model()`, which simplifies model loading and inference, making deep learning techniques accessible to users without requiring specialized expertise in computer vision.

## Statement of Need

The accurate and timely identification of bark and ambrosia beetle species is fundamental to forest health monitoring, pest management, and quarantine efforts worldwide (Hulcr & Stelinski, 2017; Piper et al., 2019; Ramsfield et al., 2016). Many species are cryptic or morphologically similar (i.e., exhibit high homoplasy), making identification a significant bottleneck that requires highly specialized taxonomic expertise (Hulcr et al., 2015; Kirkendall et al., 2015). Traditional morphological identification methods are slow, not scalable for large ecological datasets, and can be prohibitively expensive, hindering rapid responses to invasive species outbreaks and broad-scale biodiversity research.

While computer vision offers a promising solution, a critical gap exists in the available software ecosystem. There is a lack of specialized, accessible, and benchmarked tools for bark beetle identification. `ibbi` fills this gap by providing a robust, easy-to-install Python package that:

1. Focuses specifically on the taxonomically challenging and economically critical bark and ambrosia beetle groups.
2. Offers a range of pre-trained, benchmarked models for different tasks (object-detection and object-classification, zero-shot object-detection).
3. Standardizes model access through a simple, high-level API, lowering the barrier to entry for researchers.

By consolidating these functionalities into a single package, `ibbi` provides a foundational tool that enables reproducible, high-throughput research that was previously infeasible.

## State of the Field

Automated species identification using computer vision has seen significant progress, with deep learning models achieving expert-level accuracy in various biological domains ([Høye et al., 2021](#); [Valan et al., 2019](#)). This progress has been largely driven by two trends: broad-scale generalist models and bespoke academic models.

Large-scale projects, often powered by citizen-science datasets like iNaturalist, have successfully developed generalist models for identifying a wide range of common taxa ([Horn et al., 2018](#)). Similarly, comprehensive toolkits like Pytorch-Wildlife offer powerful, pre-trained models, such as MegaDetector, for identifying a broad spectrum of wildlife in camera trap images ([Hernandez et al., 2024](#)). However, these generalist models, while powerful for common categories, often struggle with the fine-grained distinctions required for less common or cryptic species, particularly in specialized groups like bark and ambrosia beetles, where morphological similarities can lead to misidentification ([Kirkendall et al., 2015](#)).

This has led to a growing trend of developing specialized tools for specific taxonomic groups of high economic or ecological importance. For example, various research initiatives have focused on creating models for identifying bees to monitor pollinators or for classifying mosquito species to track disease vectors ([Buschbacher et al., 2020](#); [Goodwin et al., 2021](#)). These efforts demonstrate the recognized need for domain-specific tools, but the resulting models are rarely distributed in an accessible, unified software package.

On the other end of the spectrum, numerous studies have developed bespoke computer vision models for bark beetle classification ([Marais et al., 2025](#); [Sun et al., 2024](#)). While valuable, these models are typically created for a specific research question and are not released as accessible, reusable, or maintained software. Consequently, they are not universally benchmarked, and their performance on different datasets is unknown.

`ibbi` is novel in that it bridges this gap. It is, to our knowledge, the first software package to:

1. Aggregate multiple deep learning architectures specifically for bark and ambrosia beetle identification into a single, cohesive framework.
2. Train these models on one of the largest and most taxonomically diverse datasets of bark and ambrosia beetles available.
3. Provide a standardized method for benchmarking these models, allowing for direct performance comparisons.
4. Distribute the models and inference code as an easy-to-install Python package, promoting transparency, reusability, and extensibility.

The first version of `ibbi` is made possible by powerful open-source libraries that provide state-of-the-art model implementations, such as timm ([Wightman, 2019](#)) and ultralytics ([Jocher et al., 2020](#)), and frameworks such as Hugging Face for model and data sharing ([Wolf et al., 2019](#)).

Consequently, `ibbi` is established not as a substitute for broad frameworks like MegaDetector, but as a specialized complement that fills a critical gap in functionality, while simultaneously overcoming the accessibility barriers often associated with bespoke academic models.

## Functionality

The `ibbi` package is designed for ease of use and flexibility. Its core functionality revolves around a factory function, `create_model()`, which handles model instantiation and the downloading of pre-trained weights from Hugging Face Hub ([Wolf et al., 2019](#)).

## Model Curation and Datasets

The models provided in `ibbi` are the result of a comprehensive data pipeline (Figure 1). The process involved collecting images from diverse sources, using a zero-shot detection model for initial beetle localization, and employing a human-in-the-loop process to verify annotations. The curated data was then used to train the object-detection and object-classification models. To promote transparency and reproducibility, the package provides a helper function to access the testing dataset used for evaluation. Furthermore, users can programmatically view all available models and their performance metrics using the `ibbi.list_models()` function, allowing for informed model selection.

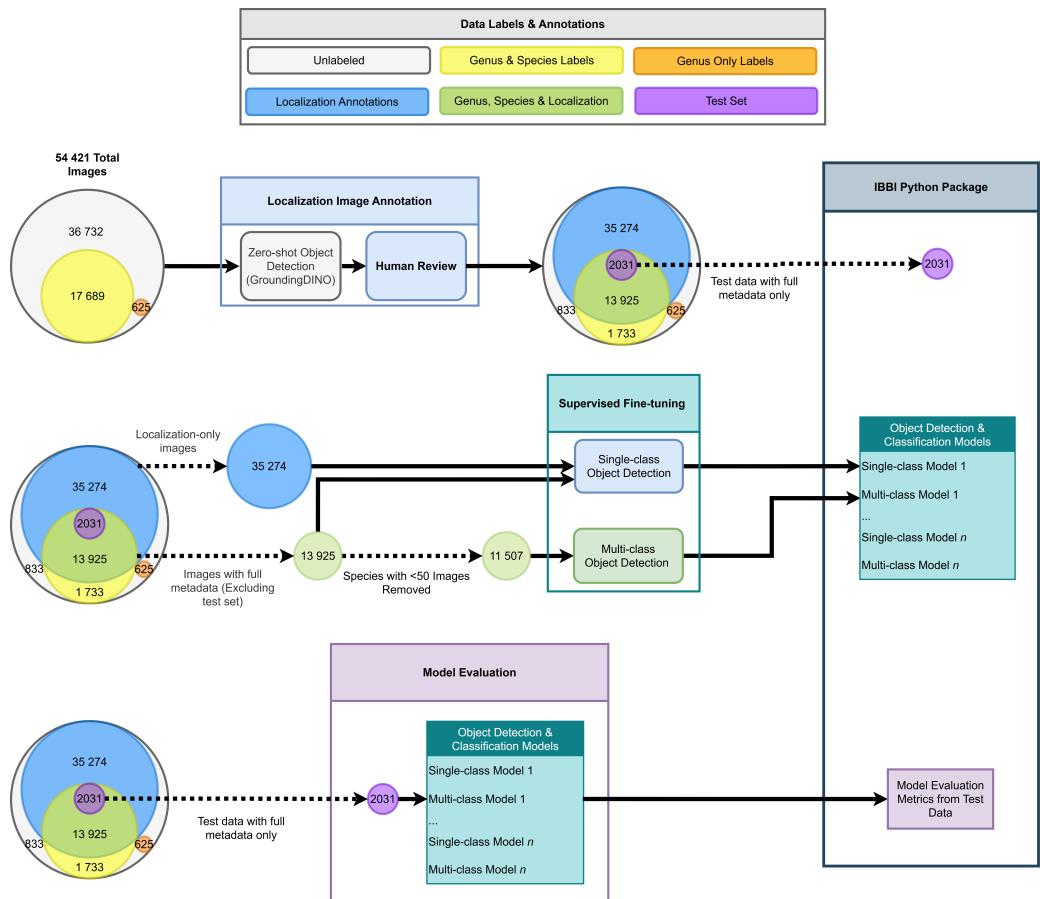


Figure 1: Overview of the data collection, curation, and model training workflow for the `ibbi` package.

## Core Tasks and API

The package supports four primary tasks:

- **Single-class Object-Detection:** Identifies and localizes any bark and ambrosia beetles in an image, returning bounding box coordinates. This is ideal for processing raw images from traps or collection sheets and is based on fine-tuned high-performance architectures like YOLO (Redmon et al., 2016) and detection transformers (Lv et al., 2023).
- **Zero-Shot Object-Detection:** Detects arbitrary objects in an image based on a user-provided text prompt. This powerful feature uses the GroundingDINO model (Liu et al., 2023) and allows for flexible analysis beyond predefined categories. However, this model has not been fine-tuned specifically on bark and ambrosia beetle images.
- **Multi-class Object-Detection:** Predicts the species of a beetle from an image, returning

species class probabilities. This currently supports 63 species and is also based on fine-tuned high-performance architectures like YOLO (Redmon et al., 2016) and detection transformers (Lv et al., 2023). However, it is important to note that the models are not fine tuned on a dataset as large as the one used for single-class object-detection, and therefore may not perform as well on images that are not tightly cropped around the specimen. This task is the main focus for the future roadmap of the package, with plans to expand the number of species supported and improve model performance.

- **Feature Extraction:** Generates deep-learning feature embeddings (vectors) from an image. This task is available for models of all tasks. These feature embeddings can be used for downstream tasks like clustering, similarity analysis, or visual search engines.

#### A typical Workflow:

```
import ibbi
from PIL import Image

# 1. List available models to see choices and performance metrics
ibbi.list_models()

# 2. Choose and load a pre-trained model by name
# For this example, we'll use a species detection and classification model
model_name = "yolov10x_bb_multi_class_detect_model"
model = ibbi.create_model(model_name, pretrained=True)

# 3. Prepare an image (from path, URL, or PIL/Numpy object)
image = Image.open("path/to/beetle_image.jpg")

# 4. Run inference
results = model.predict(image)

# 5. Extract features
features = model.extract_features(image)
```

#### Inputs, Outputs, and Hardware

- **Inputs:** The predict and extract\_features methods accept image inputs as a file path, a URL, a PIL.Image object, or a numpy.ndarray.
- **Outputs:** The predict method returns a list of custom Results objects, which contain easily accessible attributes such as bounding boxes (.boxes), probability scores (.probs), and class names (.names).
- **Hardware & Limitations:** While ibbi can run on a standard CPU, a CUDA-enabled GPU is highly recommended for faster inference, especially for batch processing. The performance of the models is dependent on the quality and resolution of the input images. Object-classification models achieve the highest accuracy when used on images that are well-lit and tightly cropped around the specimen.

## Future Development Roadmap

ibbi is under active development with a commitment to long-term support and expansion. The future roadmap is centered on two key areas: expanding the core platform and developing a novel, specialized identification model.

## Platform and Ecosystem Expansion

- **Model repository expansion:** We are committed to continually expanding the ibbi model repository by fine-tuning additional state-of-the-art architectures on our dataset.
- **Dataset Collection Growth:** The training dataset itself will be perpetually augmented with more images and species to improve taxonomic coverage.
- **Enhanced Evaluation Framework:** We will expand the evaluation capabilities of the package to serve as a standardized benchmarking tool for other models in the field. This will include integrating explainable AI (XAI) techniques for better model interpretability and enhancing the ability to perform clustering analyses on feature embeddings.
- **Specialized Identification Model Development:** We will develop a new, specialized identification model that is optimized for bark beetle identification, moving beyond the generalist models currently available. This model will be modular and incrementally developed, with new capabilities released through package updates.

## Licensing, Authorship, and Conflicts of Interest

### License

This software is licensed under the OSI-approved MIT License.

### Authorship

All authors have made substantial contributions to the design of the software.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgements

We thank the numerous collaborators, students, and technicians in the Hulcr lab at the University of Florida who have contributed to the collection, curation, and annotation of the image dataset that made this work possible. We also acknowledge the National Science Foundation, the USDA Forest Service, and the Florida Forest Service for their financial support.

## References

- Buschbacher, K., Ahrens, D., Espeland, M., & Steinhage, V. (2020). Image-based species identification of wild bees using convolutional neural networks. *Ecological Informatics*, 55, 101017. <https://doi.org/10.1016/j.ecoinf.2019.101017>
- Goodwin, A., Padmanabhan, S., Hira, S., Glancey, M., Slinowsky, M., Immidisetti, R., Scavo, L., Brey, J., Sai Sudhakar, B. M. M., Ford, T., & others. (2021). Mosquito species identification using convolutional neural networks with a multitiered ensemble model for novel species detection. *Scientific Reports*, 11(1), 13656. <https://doi.org/10.1038/s41598-021-92891-9>
- Hernandez, A., Miao, Z., Vargas, L., Beery, S., Dodhia, R., Arbelaez, P., & Ferres, J. M. L. (2024). Pytorch-wildlife: A collaborative deep learning framework for conservation. *arXiv Preprint arXiv:2405.12930*. <https://doi.org/10.48550/arXiv.2405.12930>
- Horn, G. V., Aodha, O. M., Song, Y., Shepard, A., Adam, H., Perona, P., & Belongie, S. (2018). The iNaturalist species classification and detection dataset. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8769–8778. <https://doi.org/10.1109/CVPR.2018.00914>

- Høye, T. T., Ärje, J., Bjerge, K., Hansen, O. L. P., Iosifidis, A., Leitch, K. J., Mann, H. M. R., Meissner, J. J., Melvad, C., & Raitoharju, J. (2021). Deep learning and computer vision will transform entomology. *Proceedings of the National Academy of Sciences*, 118(2), e2002545117. <https://doi.org/10.1073/pnas.2002545117>
- Hulcr, J., Atkinson, T. H., Cognato, A. I., Jordal, B. H., & McKenna, D. D. (2015). Morphology, taxonomy, and phylogenetics of bark beetles. In F. E. Vega & R. W. Hofstetter (Eds.), *Bark beetles: Biology and ecology of native and invasive species* (pp. 41–84). Academic Press. <https://doi.org/10.1016/B978-0-12-417156-5.00002-2>
- Hulcr, J., & Stelinski, L. L. (2017). The ambrosia symbiosis: From evolutionary ecology to practical management. *Annual Review of Entomology*, 62, 285–303. <https://doi.org/10.1146/annurev-ento-031616-035105>
- Jocher, G., Stoken, A., Borovec, J., & others. (2020). Ultralytics YOLOv5. <https://github.com/ultralytics/yolov5>; Zenodo. <https://doi.org/10.5281/zenodo.3908559>
- Kirkendall, L. R., Biedermann, P. H. W., & Jordal, B. H. (2015). Evolution and diversity of bark and ambrosia beetles. In F. E. Vega & R. W. Hofstetter (Eds.), *Bark beetles: Biology and ecology of native and invasive species* (pp. 85–156). Academic Press. <https://doi.org/10.1016/B978-0-12-417156-5.00003-4>
- Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Su, J., Zhu, J., Zhang, L., & Su, H. (2023). Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14480–14490. <https://doi.org/10.1109/ICCV51070.2023.01349>
- Lv, W., Gao, S.-H., He, C., Jiao, J., Wang, S., Liu, Z., Wang, Y., Wang, L., Han, K., Chen, C., You, Y., & Wang, J. (2023). DETRs with Collaborative Hybrid Assignments for Real-Time Object Detection. *arXiv Preprint arXiv:2304.08069*. <https://doi.org/10.48550/arXiv.2304.08069>
- Marais, G. C., Stratton, I. C., Johnson, A. J., & Hulcr, J. (2025). Progress in developing a bark beetle identification tool. *PLoS One*, 20(6), e0310716. <https://doi.org/10.1371/journal.pone.0310716>
- Piper, A. M., Batovska, J., Cogan, N. O. I., Cunningham, J. P., & Blacket, M. J. (2019). Prospects and challenges of implementing DNA metabarcoding for high-throughput insect surveillance. *GigaScience*, 8(8), giz092. <https://doi.org/10.1093/gigascience/giz092>
- Ramsfield, T. D., Bentz, B. J., Faccoli, M., Jactel, H., & Brockerhoff, E. G. (2016). Forest health in a changing world: Effects of globalization and climate change on forest insect and pathogen impacts. *Forestry: An International Journal of Forest Research*, 89(3), 245–252. <https://doi.org/10.1093/forestry/cpw018>
- 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*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Sun, L., Cai, Z., Liang, K., Wang, Y., Zeng, W., & Yan, X. (2024). An intelligent system for high-density small target pest identification and infestation level determination based on an improved YOLOv5 model. *Expert Systems with Applications*, 239, 122190. <https://doi.org/10.1016/j.eswa.2023.122190>
- Valan, M., Makonyi, K., Maki, A., Vondráček, D., & Ronquist, F. (2019). Automated taxonomic identification of insects with expert-level accuracy using effective feature transfer from convolutional networks. *Systematic Biology*, 68(6), 876–895. <https://doi.org/10.1093/sysbio/syz014>
- Wightman, R. (2019). PyTorch image models. In *GitHub repository*. <https://github.com/rwightman/pytorch-image-models>; GitHub. <https://doi.org/10.5281/zenodo.4414861>

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & others. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv Preprint arXiv:1910.03771*. <https://doi.org/10.48550/arXiv.1910.03771>