BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Beetle tracking

Bc. Dalibor Křička*, Bc. Jakub Pekárek**, Bc. Pavel Osinek***

Abstract

This work addresses the problem of tracking many beetles in top-down view video recordings, specifically Tenebrio molitor species. The objective was to develop a tool for long-term tracking of individual beetles, emphasizing maintaining consistent identities across frames. An annotated dataset was prepared to support this goal, and a YOLOv11m detection model was fine-tuned. The system integrates the ByteTrack and BoT-SORT tracking algorithms, which were selected for efficiency and accuracy. The resulting solution enables automated beetle movement and behavior analysis, providing valuable data for biological research based on video recordings.

*xkrick01@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*
**xpekar19@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*
***xosine00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

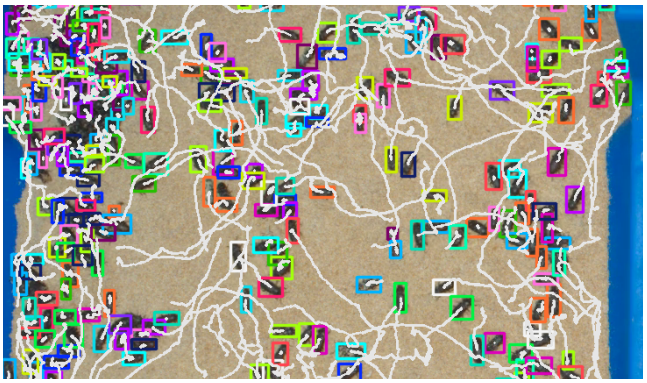

Figure 1. Example of a trajectory of currently detected and simultaneously identified beetles after 300 recording frames.

1. Introduction

Understanding insect behavior is crucial for various biological disciplines, including ethology and pest management. Accurate and continuous tracking of individual beetles over extended video sequences enables detailed analysis of their activity, movement patterns, and interactions. However, manual annotation of such data is time-consuming and prone to errors, highlighting the need for automated tracking systems.

This work addresses a long-term multi-object tracking (MOT) challenge focused explicitly on beetles (Tenebrio molitor species). The goal is to obtain trajectories of individual beetles while preserving their identities throughout the video.

2. Related Work

In recent years, there has been significant progress in multi-object tracking (MOT), particularly driven by advances in deep learning methods [1, 2]. Nevertheless, classical tracking algorithms such as SORT remain popular due to their simplicity and speed. It's extension, DeepSORT [3], incorporates appearance features into the object association process, significantly improving the distinction between individual tracked objects and reducing identity switches.

2.1 Tracking Algorithms

Recently, increasing attention has been given to newer methods that more effectively handle challenging scenarios with high object density. Notable approaches include ByteTrack [4], BoT-SORT [5], BoostTrack [6], DeepOCSORT [7], and ImprAssoc [8]. ByteTrack enhances the association stage by including low-confidence detections, which improves tracking under partial occlusions. BoT-SORT combines a motion model with appearance features and camera motion compensation. BoostTrack introduces adaptive weighting based on detection confidence and tracklet stability. DeepOCSORT employs dynamically weighted appearance features and precise motion compensation. ImprAssoc unifies the association of all detections by integrating appearance and motion cues while considering occlusions during the initialization of new tracks.

2.2 Object Detectors

The choice of object detector heavily influences tracking quality. Models from the YOLO (You Only Look Once) family are most commonly used due to their high speed and strong accuracy, making them suitable for real-time applications.

Recent versions, such as YOLOv11 and YOLOv12, offer significant improvements in detection accuracy and efficiency. YOLOv12, in particular, emphasizes using attention mechanisms further to enhance real-time detection performance [9, 10]. In addition to YOLO, several other well-established approaches exist:

SSD (Single Shot Detector)[11] achieves high accuracy while maintaining speed by predicting bounding boxes and object classes in a single pass through the network across multiple feature maps.

RetinaNet[12] addresses the class imbalance problem between foreground and background in dense detectors using the so-called Focal Loss.

Faster R-CNN [13] introduced the Region Proposal Network (RPN), which significantly accelerated region proposal generation, making the detection process faster than earlier R-CNN models.

These models offer various trade-offs between speed and accuracy and are selected based on the specific requirements of the tracking application.

3. Methodology

3.1 Dataset

The dataset consists of eight video recordings with varying beetle population densities. These recordings cover four levels of beetle density in the scene:

- density 1: 0.15 beetles/cm^2;
- density 2: 0.25 beetles/cm^2;
- density 3: 0.375 beetles/cm^2;
- density 4: 0.425 beetles/cm^2.

Initial annotations were generated automatically using grayscale thresholding, allowing potential objects to be identified via contour detection. The resulting bounding boxes were then manually revised – erroneous boxes were removed, missing annotations for undetected objects were added, and corrections were made in cases where a bounding box included multiple beetles or otherwise incorrectly delineated object boundaries.

In the second phase, a YOLO model was fine-tuned on the manually annotated data, enabling more accurate automatic annotation of additional video segments. This resulted in a significantly larger volume of annotated data with minimal need for manual correction.

A partial *ground truth* was manually created to evaluate specific tracking metrics – specifically, 45 trajectories of individual beetles, each covering 150 consecutive frames. The trajectories were selected to represent a range of tracking difficulties. Challenging trajectories included, for instance, cases where the beetle moved through densely populated areas or frequently crossed paths with other beetles. In contrast, simpler trajectories featured beetles moving in isolation and at a sufficient distance from others.

This approach allowed for meaningful and objective evaluation of tracking performance in an environment where complete annotation of all objects is not feasible; see section 3.4.

3.2 Detection

The YOLOv11m model was fine-tuned on the described dataset with the following configuration:

- training images: 79;
  - 14,319 objects;
- validation images: 20;
  - 3,357 objects;
- image resolution: 640×640 px;
- number of epochs: 100;
- batch size: 8;
- maximum number of detections: 1000;
- total number of annotated objects: 17,676.

The dataset was created from 640×640 px image patches randomly sampled from various videos, frames, and positions within those frames. This approach ensured that the small beetles (approximately 30 px in length and around 15 px in width) were not further downscaled during training, as the model operated on cropped patches at their original resolution from the source data.

The best model trained on this dataset achieved the following results:

- mAP@0.5: 0.983;
- recall: 0.951;
- precision: 0.965;
- inference time: 58.7 ms.

The training quality was monitored using precision-recall curves, confusion matrices, and validation metrics.

The training was also tested without using the built-in data augmentation. However, models trained in this way achieved significantly worse metrics — for instance, mAP@0.5 dropped below 0.1 — and this direction was not pursued further. The underlying

hypothesis was that augmentation might not be necessary if the final model were to be deployed only on very similar data — that is, the same scene with different arrangements of individuals of the same species. Partial disabling of specific augmentation types, such as translation, scaling, mosaic augmentation, hue adjustment, or cutout masking, was also tested. However, these modifications led to decreases in mAP@0.5 and other performance metrics.

A significant breakthrough during the initial training was discovering that the default setting for the maximum number of detections was 300. This meant that only up to 300 beetles were detected during validation, even though more than 500 beetles were present in the entire image, even at the lowest density level. After increasing this parameter to 1000, all performance metrics significantly improved. For example, mAP@0.5 increased from below 0.1 to more than 0.4.

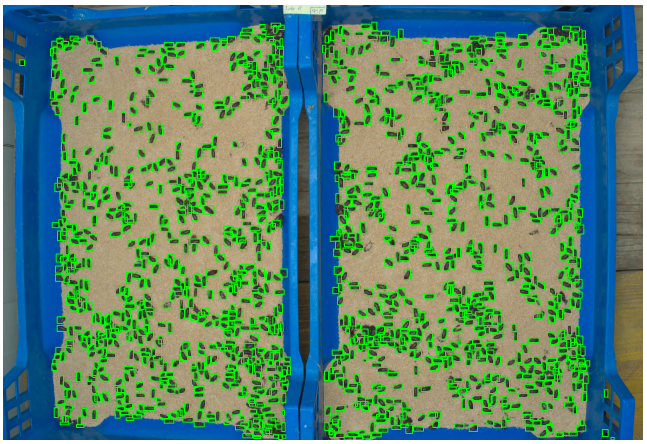

Figure 2. Beetles detected by the model with input size 640×640 px. The image is taken from the video with the highest beetle density. One thousand four hundred ninety-four beetles were detected, and the inference time was 82.6 ms.

Figure 2 shows the detections made by the best-performing model with an input size of 640×640 px. During detection, the IoU threshold was set to 0.5 and the confidence threshold to 0.4. Increasing the confidence threshold reduces the number of falsely detected bounding boxes near the edges of actual beetles but also decreases the number of correctly detected individuals. These threshold values were, therefore, selected as the best trade-off between precision and recall.

In addition, detection performance decreases in regions with high beetle density. This is caused by the limited video quality, making distinguishing individual beetles in densely populated areas difficult.

Figure 3. Beetles that are difficult to detect.

As illustrated in Figure 3, even manual annotation would struggle to mark all beetles in such scenarios correctly.

3.3 Tracking

Object tracking involves the consistent assignment of identifiers to detected objects across consecutive frames, allowing the reconstruction of their trajectories over time. Given the high number of objects in the scene and the absence of complete ground truth data, it was essential to use algorithms capable of operating robustly, even in the presence of occasional detection failures or object occlusions. For this task, the ByteTrack and BoT-SORT algorithms were selected.

The tracking algorithms were applied to the detection outputs generated by the fine-tuned YOLOv11m model (see Section 3.2). The trackers were not further trained or modified and were used in their original implementation with default parameters. All tracking methods were evaluated using the same input — a sequence of bounding boxes with associated confidence scores and timestamps — to ensure fair comparison.

As the performance of a tracker strongly depends on the quality of the input detections, failures in detection (e.g., in dense clusters or under partial occlusion) directly affect the stability of the tracking. This issue is further analyzed in the following section, which focuses on evaluating tracking quality.

3.4 Evaluation of Tracking Quality

To evaluate the quality of individual beetle tracking, we selected a set of metrics focused on identity consistency, known as identity-based tracking metrics. Given the nature and scale of our dataset—eight videos, each 1,000 frames long, with between 500 and 1,500 beetles per video—it was not feasible to manually annotate a complete ground truth (GT) for all trajectories. Therefore, we opted not to use traditional metrics such as MOTA (Multiple Object Tracking Accuracy) or HOTA (Higher Order Tracking Accuracy), which require complete and continuous annotations of all objects in the scene. Under conditions of incomplete ground truth, such metrics would not provide reliable results.

Instead, we employed metrics suitable for partially annotated data, which focus on the tracker's ability to maintain correct object identities or determine whether identity switches were caused by the inability to detect the object properly (e.g., within clusters of multiple beetles). Table 1 presents the selected metrics and their descriptions. To compute these metrics, we used the ground truth for 45 individual beetle trajectories, which are described in more detail in the dataset chapter (see Section 3.1).

Metric	Description
IDF1	Harmonic mean of IDPrec and IDRec.
IDPrec	Proportion of correctly assigned IDs among all detections.
IDRec	Proportion of correctly detected and identified beetles out of all ground truth identities.
IDTP	Number of correctly assigned IDs to detections.
IDFP	Number of incorrectly assigned IDs to detections.
IDFN	Number of frames in which a beetle was not detected.
Frag	Number of times a single ground truth trajectory was fragmented into multiple segments.
IDSwitch	Number of identity switches during the tracking of a single object.
FSRatio	Ratio of ID switches caused by detection failures (Frag) to the total number of ID switches on a trajectory (IDSwitch).
meanLen	Average length of correctly tracked trajectory segments (maximum 150).

Table 1. Metrics and their descriptions used for evaluating individual tracking algorithms.

3.5 Tracking Quality Results for Beetles

Individual object (beetle) tracking quality was analyzed using BoT-SORT and ByteTrack tracking algorithms. The performance metric values introduced in the previous section are summarized in Table 2.

Based on the results of the *IDF1* metric, which reflects the algorithm's ability to assign correct identities to objects across frames consistently, it can be concluded that the differences between the evaluated approaches are not substantial.

Figure 4 visualizes the trajectories for which at least one of the trackers exhibited a low IDF1 score. While some cases reveal that a specific algorithm performed better than the others, no method consistently outperformed the rest across all scenarios. The performance of each algorithm varies depending on the trajectory being tracked, and in problematic cases, all methods frequently fail.

Another perspective is provided by the *FSRatio* metric, which quantifies the proportion of identity switches caused by detection loss relative to the total number of identity switches. The results suggest that significant performance degradation in the trackers is often linked to the detector's limited ability to recognize objects under challenging conditions.

From these observations, it can be inferred that the shortcomings in object tracking are primarily due to detection-related issues rather than the tracking mechanism itself (see Section 3.2). An example of a trajectory interruption caused by faulty detection is shown in Figure 5.

Tracker	IDF1	IDPrec	IDRec	IDTP	IDFP	IDFN	Frag	IDSwitch	FSRatio	meanLen
BoT-SORT	0.9109	0.9055	0.9164	5644	589	515	23	25	0.92	130.87
ByteTrack	0.8967	0.9051	0.8884	5484	575	689	27	30	0.90	130.60

Table 2. Summary of performance metrics for individual tracking algorithms (aggregated across all trajectories).

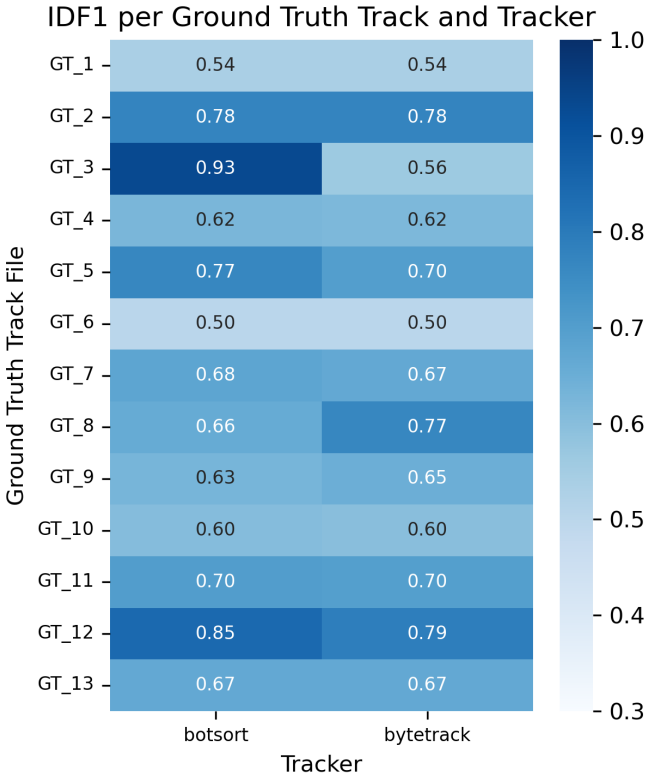

Figure 4. IDF1 scores for problematic ground truth trajectories across different tracking algorithms. No algorithm demonstrates a superior performance.

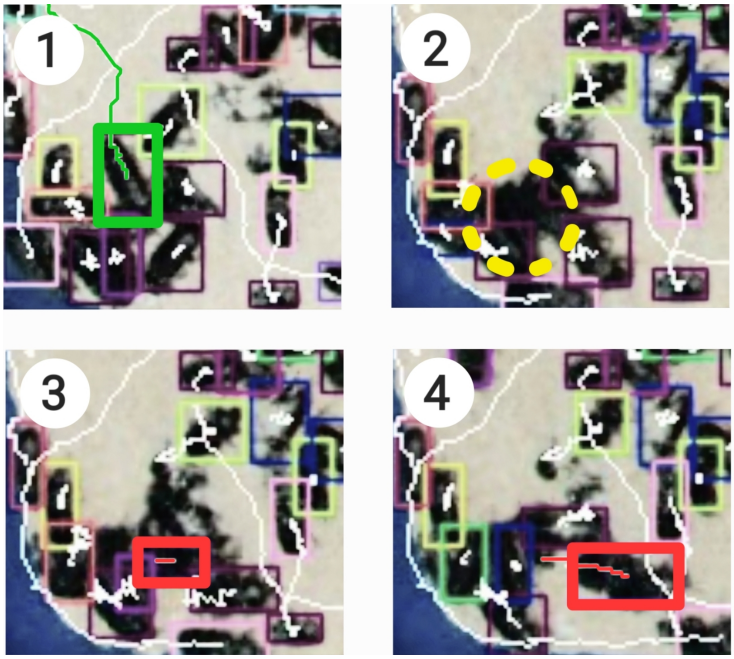

Figure 5. Loss of beetle trajectory due to detection failure in a cluster. Image 1 shows the original trajectory. Image 2 illustrates the detector's failure to recognize the beetle across several consecutive frames. Images 3 and 4 depict the re-detection of the same beetle, now assigned a new identity.

4. Conclusion

This work aimed to design and evaluate a system for detecting and tracking many beetles in video recordings with varying object densities. In the first stage, the YOLOv11m detection model was successfully fine-tuned. With appropriate settings (e.g., increasing the maximum number of detections), the model achieved high accuracy, reaching up to mAP@0.5 = 0.983. However, specific challenges remain, particularly in regions with high beetle density, where limited video resolution and object occlusions reduce detection quality.

Two multi-object tracking algorithms subsequently processed the detection outputs. The results indicated that tracking is relatively robust, and the performance differences between the selected trackers were not substantial. The key factor influencing tracking performance was the reliability of detections — detection failures were the primary cause of identity switches and trajectory interruptions.

These findings highlight the need for improvements, primarily in the detection stage. This could include better video preprocessing, training on higher-quality datasets, or deploying detectors with greater robustness to noise and compression artifacts. Future work should also expand the manually annotated ground truth and evaluate tracker performance using established metrics such as MOTA or HOTA.

References

[1] Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019.

[2] Patrick Dendorfer, Aljosa Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking. *International Journal of Computer Vision*, 129:845–881, 2021.

[3] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. *CoRR*, abs/1703.07402, 2017.

[4] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, and Ping Luo. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. In *ECCV*, 2022.

[5] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking, 2022.

[6] Longteng Kong, Qiankun Ding, Nan Xue, and Yunde Jia. BoostTrack: Towards High Performance Online Multi-Object Tracking. *arXiv preprint arXiv:2303.09599*, 2023.

[7] Eugenio Bochinski, Thomas Senst, Timo Fischer, and Thomas Sikora. Deep OC-SORT: Advanced Track Association Strategies for High Performance Online Multi-Object Tracking. *arXiv preprint arXiv:2304.00962*, 2023.

[8] Daniel Stadler and Jürgen Beyerer. An improved association pipeline for multi-person tracking. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3170–3179, 2023.

[9] Yunjie Tian, Qixiang Ye, and David Doermann. Yolov12: Attention-centric real-time object detectors, 2025.

[10] Rahima Khanam and Muhammad Hussain. A review of yolov12: Attention-based enhancements vs. previous versions, 2025.

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. *SSD: Single Shot MultiBox Detector*, page 21–37. Springer International Publishing, 2016.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.