A Comparative Analysis of Tracking Algorithms for Hornet Monitoring Using Custom-trained YOLO (You Only Look Once) Model

Abstract

This study focused on training the YOLO object detection model using a custom dataset tailored specifically for detecting hornets in and around their nests. Subsequently, the trained model was utilized to conduct a comparative evaluation of three distinct tracking algorithms—BoTSort, ByteTrack, and DeepSort—to identify the most effective one for tracking hornets in real-world scenarios. The evaluation included assessing each tracking algorithm's performance on an unseen test video sequence, considering critical factors such as detection accuracy, precision, the ability to maintain object identities across frames, as well as CLEAR metrics such as MOTA (Multiple Object Tracking Accuracy) and MOTP (Multiple Object Tracking Precision). The findings of this study indicate that both BoTSort and ByteTrack demonstrated superior effectiveness in accurately tracking hornets within video sequences compared to DeepSort. Among these tracking algorithms, ByteTrack stood out due to its lower incidence of ID switches, making it the recommended tracking algorithm for hornet monitoring.

Introduction

The Asian Hornet species, Vespa velutina, distinguished by its characteristic yellow legs (Fig. 1), was inadvertently introduced to Europe from Southern China. Initially observed in France in 2004, these hornets have since proliferated across Europe, notably in countries such as France, Italy, Spain, and Portugal (Ueno, T., 2014). They pose a significant threat to native honeybee populations by preying on them, resulting in economic losses for beekeepers and ecological harm to biodiversity. Despite their successful invasion, much of their biology and behavior remains elusive (Monceau et al., 2014), partly due to the challenges associated with tracking their movements owing to their diminutive size (Lioy, S. et al., 2021).

Research Objective

Machine learning presents a promising approach for detecting and monitoring these hornets at scale, thereby assisting in containment efforts. This research aims to advance Asian Hornet detection by enhancing training datasets with accurately labeled images extracted from video frames. Additionally, the study will conduct a comparative analysis of three state-of-the-art multiple objective trackers—BoT-SORT, ByteTrack, and DeepSORT—evaluating them using CLEAR (Leal, T. et al., 2017) metrics such as Multiple Object Tracking Accuracy (MOT-A), Multiple Object Tracking Precision (MOT-P), accuracy, precision, and recall. The primary objective is to identify the most suitable algorithm among these trackers for Asian Hornet monitoring and containment.

Fig.1 A worker of the invasive hornet Vespa velutina collected in Tsushima Island (Ueno, T., 2014)

Background

1. Multiple Object Detection Algorithm

The evolution and impact of Yolo (You Only Look Once) in computer vision for real-time object detection have been transformative since its inception in 2015. Prior to Yolo, algorithms such as R-CNN, Fast R-CNN, and Faster R-CNN dominated the field. These earlier approaches relied on a two-stage detection process involving region proposal networks and subsequent classification and bounding box refinement.

In contrast, Yolo introduced a revolutionary single-stage approach that significantly improved speed and efficiency. By leveraging Convolutional Neural Networks (CNNs) for feature learning and prediction, Yolo streamlined the object detection pipeline into a single pass through the network (Sultana, F., 2020). This integration of feature extraction, bounding box regression, and object classification within one unified framework dramatically reduced computational complexity and enabled near real-time performance.

The key steps of the Yolo algorithm as illustrated in Fig. 2 are as follows:

a. **Input Image Processing**: The input image is resized to a fixed size (e.g., 64x64) for consistent processing.
b. **Feature Extraction using CNN (Convolutional Neural Networks)**: The entire image is

processed through CNN to extract relevant features capturing both low-level details (e.g., edges, textures) and high-level semantic information (e.g., object shapes, contexts).

c. **Grid Prediction**: The CNN output is a feature map divided into an SxS grid. Each grid cell predicts bounding boxes (usually represented as grid-relative coordinates) and confidence scores predicting the presence for different object classes.

d. **Non-Maximum Suppression (NMS)**: Yolo employs NMS to refine predictions by filtering out redundant bounding boxes with overlapping regions and lower confidence scores, retaining only the most confident and non-overlapping predictions.

e. **Output**: The final output comprises bounding boxes with associated class labels, bounding boxes with confidence score reflecting the model's prediction certainty.

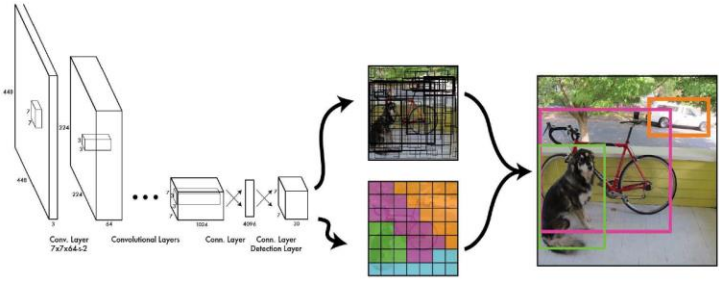

Fig. 2 You only look once (YOLO) model

Tracking Algorithms Evaluated

The objective of multi-object tracking, illustrated by scenarios such as tracking multiple objects within a video frame, is to detect and continuously monitor each object, assigning a consistent identifier to maintain continuity across successive video frames. As objects change position or become temporarily occluded, a robust tracking algorithm updates their positions and retains assigned IDs. The tracking process involves three main steps: detection, motion prediction, and data association (Zhang, Y. et al., 2022).

Detection: Initially, objects are detected and initialized into trajectories or tracklets in the first frame.

Motion Prediction: Using motion models such as the Kalman filter, the predicted locations of tracklets are estimated for the subsequent frame.

Data Association: In the following frame, detected objects are associated with predicted tracklet locations to identify and maintain object identities.

Through these steps, multi-object tracking aims to accurately follow objects over time despite changes in appearance, occlusion, or movement.

This study assesses three tracking algorithms based on their performance, specifically when integrated with the YOLO model trained to detect Hornets. The tracking algorithms considered are as follows:

A. BoT-Sort (Bounding Box Tracker-Sort)

BoT-Sort employs a combination of the Kalman filter and the Hungarian algorithm to track detections. Moreover, it is the default tracker in YOLOv8 models developed by *'ultralytics'*, as used in this study, and is recognized as a state-of-the-art tracker, exhibiting top performance on MOT Challenge datasets such as MOT 17 and MOT20 (Aharon, N., 2022).

B. ByteTrack

ByteTrack is an innovative tracking algorithm invented by Zhang, Y., et al. that has garnered attention for its exceptional performance in handling scenarios with complex object interactions and occlusions. Unlike conventional methods, ByteTrack implements a unique data association strategy that prioritizes high-confidence detections by maintaining a hierarchical structure of detections based on their scores.

In the initial frame of a video sequence, all detected bounding boxes are initialized as tracklets. In subsequent frames, the algorithm associates high-confidence (high-score) detection boxes with these tracklets by computing spatial distance metrics such as Intersection over Union (IOU) between the high-score detections and the predicted bounding boxes of the existing tracks. This association is typically performed using the Hungarian algorithm to optimize the matching based on similarity.

During this process, some tracklets and detections may remain unmatched, often due to occlusion, motion blur, or changes in object size. Subsequently, the algorithm matches low-confidence (low-score) detection boxes with any remaining unmatched tracklets to recover objects from these low-confidence detections while discarding the rest as background noise after a predefined number of frames. Moreover, unmatched high-confidence detections are treated as new tracks to maintain continuity in object tracking (Zhang, Y. et al., 2022).

ByteTrack has demonstrated effectiveness in recovering objects from low-confidence detections, which is crucial for scenarios like Hornet tracking characterized by frequent occlusions and blurring. This is highlighted by ByteTrack ranking 1st in MOT20 challenge which contains more crowded scenes and occlusion cases (Zhang, Y. et al., 2022). This indicates that ByteTrack is robust for complex scenarios and suitable for demanding tracking scenarios such as in the case of Hornets.

C. DeepSort

DeepSort is a widely recognized algorithm developed by Wojke, N. et al. in 2017, known for its effectiveness in real-time multi-object tracking. It

leverages deep appearance features in combination with a Kalman filter for state estimation and data association (Wojke, N. et al., 2017). This approach enables the tracker to maintain object identities over longer periods of occlusion, thereby reducing the occurrence of identity switches. According to Wojke, N. et al., experimental results showed a reduction in identity switches by 45% resulting in superior tracking performance in high frame rates.

DeepSort works by employing data association techniques using a combination of a Kalman filter and the Hungarian algorithm, which computes a weighted mean of distances. Additionally, DeepSort utilizes appearance features to address challenges posed by occlusions and varying object appearances. By integrating deep learning-based appearance features with traditional state estimation and data association methods, DeepSort aims to achieve robust and accurate multi-object tracking, particularly effective in scenarios with occlusions and complex object interactions.

Methodology

The research methodology involved annotating frame-by-frame images from a video dataset to train the YOLOv8 object detection model for identifying Hornets. Subsequently, the trained model was used to predict bounding boxes for Hornets in a previously unseen test video, and three tracking algorithms—BoT-SORT, ByteTrack, and DeepSORT—were evaluated for their ability to track these predicted objects and assign unique identifiers across frames. The output of each tracker, including bounding boxes and assigned identifiers for Hornets, was recorded into CSV files for analysis. To enable accurate evaluation, a ground truth CSV file was created using the CVAT (Computer Vision Annotation Tool) tool, annotating Hornet IDs frame by frame throughout the test video. The performance of the tracking algorithms was assessed using CLEAR metrics such as Multiple Object Tracking Accuracy (MOT-A) and Multiple Object Tracking Precision (MOT-P), computed using the py-motmetrics library in Python. This evaluation facilitated comparison between the ground truth data and the tracking results, allowing for analysis of total object detections, misses, true positives, false negatives, and misidentifications of Hornet IDs and ID switches. Below describes the process in detail -

1. Data Source

The data source used for developing the Hornet object detection and tracking system comprised seven MOV or MP4 format videos, each between 1-15 minutes in duration. These videos depicted Hornet nests and the movement of Hornets in and around their habitats. The footage varied, with some videos featuring close-up views of Hornet activity, while others provided broader perspectives showing the nest and its surroundings to capture Hornets in action beyond the nest area. Notably, all videos exclusively focused on a single class of objects: Hornet.

2. Creating Custom Training Dataset

To prepare a robust dataset for training an object detection model capable of identifying Hornets in videos, best practices in computer vision for image labeling and annotation were followed. Research has emphasized that detection quality significantly impacts tracking performance, with the potential for enhancing tracking accuracy by up to 18.9% through improvements in the detector (Bewley, A. et al., 2016). Therefore, meticulous attention to dataset quality was essential.

A. Image Labelling and Annotation

The process of image labeling and annotation for Hornets was accomplished using the Computer Vision Annotation Tool (CVAT). This tool enabled precise frame-by-frame labeling, ensuring that tight bounding boxes were applied to fully encapsulate each Hornet without cutting off any part of the object. Even when Hornets were partially obscured by objects like tree branches, they were labeled as fully visible to effectively train the model in recognizing occluded instances. Consistency across images was maintained by labeling fuzzy or in-motion Hornets if identifiable by human judgment,

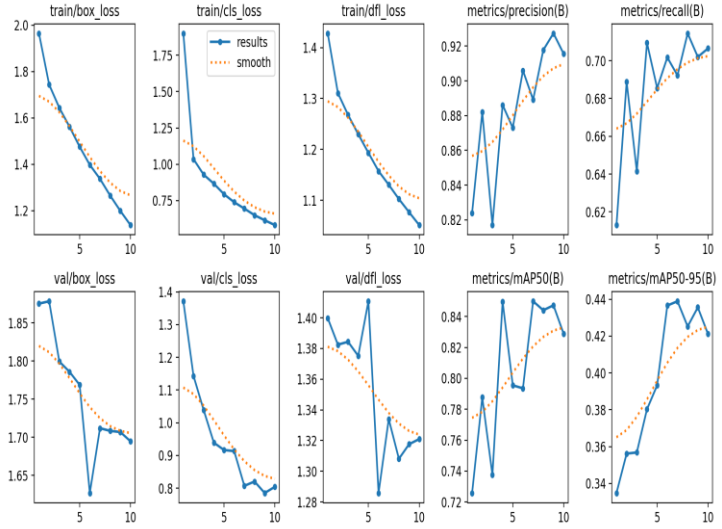

Fig. 3 Graphs showing training loss, validation loss mAP50 as well as mAP50-95 across epochs.

promoting generalization within the model. Additionally, every Hornet present in each frame was meticulously labeled to enhance model precision and minimize false negatives. The resulting annotated dataset was formatted in YOLO notation, comprising image folders containing individual frames and corresponding text files with YOLO-formatted annotations detailing object class, bounding box coordinates, and frame identifiers. This annotated dataset not only supports the training of object detection models but also facilitates transfer learning by enabling the adaptation of pre-trained models for specific tasks.

4. Model Training

The model training process utilized a dataset comprising >16,000 annotated images, which were partitioned into two distinct sets for training and validation. The training set (Y) consisted of approx. 12,000 images (60% of the total), used to train a YOLOv8n model over 10 epochs. This set enabled the model to learn object detection and classification based on the provided annotations. The validation set (W) comprised 4,000 images (20% of the total) used during training to assess the model's performance and optimize hyperparameters to prevent overfitting. Importantly, the validation set ensured that the model generalized well to new data. Additionally, a dropout of 20% of the training data was implemented as a regularization technique to mitigate overfitting, enhancing the model's ability to generalize to unseen data and improve overall performance and robustness. The decision to use 10 epochs was informed by evaluating the model's performance on the validation set, where increasing the number of epochs beyond this point resulted in diminishing returns or signs of overfitting possibly a result of small sample size as well as zoomed-out versions of miniature Hornets used in training as depicted by increasing val/dfl_loss curve towards the end in Fig.3. Despite the lower epoch, the training achieved a remarkable accuracy of 85% mAP@50 and high area under precision-recall curve as shown in Fig. 4 below.

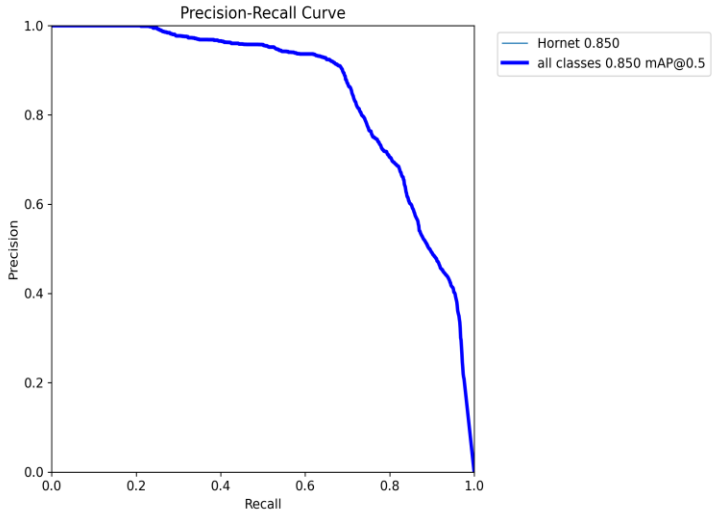

Fig. 4 Precision-Recall ROC Curve

5. Tuning Tracking Hyperparameters

Certain tracking configurations were essential for maintaining consistency and ensuring a fair evaluation of the trackers. One crucial aspect was how new detections were handled at each frame. When deciding whether a detection should be associated with an existing track or considered a new object, the following two hyperparameters were tuned across all the trackers.

A. Hornet Detection Confidence Interval

While most tracking algorithms perform well in minimizing false positives (FP), they often struggle with false negatives (FN). Research by Leal-Taixé et al. (2017) highlights that even a 10% decrease in the number of detections in the low confidence interval can lead to a 55% increase in false negatives. Baring this mind, all detections greater than 0.1 confidence intervals were sent to the trackers for detection at every frame.

B. Data Association using IOU

The data association involving Intersection over Union (IOU) threshold was set to 0.6. This threshold determines how much overlap there needs to be between a predicted bounding box and a ground truth bounding box for them to be considered a match. If a detection does not meet this IOU threshold (i.e., if the overlap is less than 60%), it is treated as a new object and assigned a new ID (Bewley, A., 2016).

Lowering the Intersection over Union (IOU) threshold to 0.6 from the default values of 0.8 for BoT-SORT and ByteTrack, and 0.7 for DeepSORT, was a deliberate strategy to increase the tolerance for matching bounding boxes, aimed at improving tracking performance by increasing the number of matched tracks and reducing the frequency of re-identification.

C. Creation and Deletion of Tracks

A critical parameter for track management was the maximum age of a track buffer, set to 900 frames, which corresponds to approximately 30 seconds at a frame rate of 30 frames per second (fps). Tracks exceeding this age are deleted prompting re-appearing objects after occlusion to be re-identified after this period. The default setting of 30 frames was too short given frequent occlusion of Hornets and therefore ignored following Bewley, A. recommendation to ignore short-term and long-term occlusion as they occur very rarely and introduce undesirable complexity into the tracking framework (Bewley, A., 2016).

6. Testing

The testing methodology employed in this research for evaluating tracking algorithms was comprehensive and involved detailed comparisons between ground truth annotations and tracking results on a frame-by-frame basis. A previously unseen test video of 7 minutes was used to perform object detection and tracking using the YOLOv8 model and tracking algorithms. Tracked objects, along with their bounding boxes, frame IDs, Hornet IDs, and class labels, were recorded and updated at each frame to maintain object identities over time.

The resulting tracking outputs, containing all relevant information, were saved into a CSV file for subsequent analysis. Ground truth annotations were meticulously created using the CVAT tool, capturing detailed

information such as frame IDs, Hornet IDs, and bounding box coordinates for each annotated Hornet instance.

To evaluate tracking performance, a range of metrics including accuracy, precision, recall, MOTA (Multiple Object Tracking Accuracy), MOTP (Multiple Object Tracking Precision), IDF1 (ID F1 Score), and IDSW (ID Switches) were calculated and analyzed. These metrics provided insights into various aspects of tracking performance, including the ability to correctly identify and track Hornets, minimize false positives and false negatives, and maintain consistent object identities over time.

The following describes the CLEAR testing metrics used to evaluate the trackers:

a. **MOTA** (Multiple Object Tracking Accuracy) - measures the overall accuracy measure by computing how often mismatches occur between tracking results and ground truth. MOTA contains the counts of False Positive (FP), False Negative (FN) and ID-switches (IDSW) normalized over the total number of ground-truth tracks (GT). Since the amount of FP and FN are larger than IDs, MOTA focuses on detection performance.

$$MOTA = 1 - \frac{\Sigma(FN + FP + IDSW)}{\Sigma GT}$$

b. **MOTP** - Average precision of object localization, calculated as the average overlap (IOU) between predicted and ground truth bounding boxes.

c. **IDF1** – This metric measures the overall performance of id-based computation by measuring how long the tracker identifies the object correctly. It is the harmonic means of identification precision (IDP) and recall (IDR).

$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN}$$

d. **IDSW** – measures when a tracker inaccurately switches the ID of the object trajectory. This is illustrated in Fig. 5 where Person A and Person B overlap in frame 4 and are not detected and tracked in frame 4-5. This results in id switches in frame 6 where person A is misidentified as person B and vice-versa. In the right side, another example shows where tracker loses track of person A from frame 4 onwards and assigned a new ID_2 when it reappears.

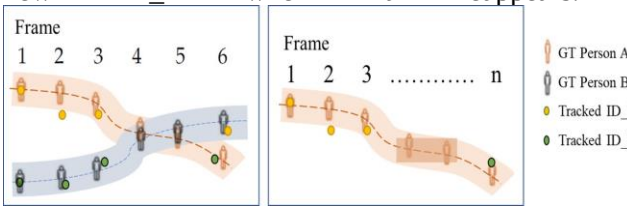

Fig. 5

By systematically comparing tracking results with ground truth annotations and assessing these performance metrics, the methodology provides a comprehensive evaluation of the effectiveness and reliability of different tracking algorithms in real-world scenarios.

Results and Discussion

The results from each tracking algorithm – BoTSort, ByteTrack and DeepSort as shown in Table 1 shows that both BoTSort and ByteTrack demonstrate comparable overall performance, with ByteTrack exhibiting slightly better IDF1 score and lesser ID switches. Both algorithms exhibit similar MOTA and MOTP, making BoTSort the practical choice due to its default integration within the YOLOv8 *'ultralytics'* package, providing a readily accessible and effective solution for Hornet tracking applications.

DeepSort, while showing stronger recall, is hindered by an extremely low accuracy of 4.5% and a higher number of identity switches as well as false positives. This discrepancy suggests potential limitations in DeepSort's long-term occlusion handling capabilities, with ByteTrack's motion models proving more reliable in such scenarios (Zhang et al., 2022).

	BoTSort	**ByteTrack**	**DeepSort**
IDF1	0.368	0.402	0.247
Recall	0.646	0.647	0.737
Precision	0.763	0.764	0.523
IDSW	82	66	594
MOTA	0.443	0.445	0.045
MOTP	0.327	0.328	0.335
False Positives	6175	6131	20680
False Negatives	10868	10859	8081

Table 1. MOT metrics comparison

Conclusion

This study focused on evaluating the performance of state-of-the-art tracking algorithms, including BoTSort, ByteTrack, and DeepSort, in effectively tracking detections identified by a custom-trained YOLOv8 model specifically designed for hornet detection in and around their nests. Through our evaluation, both BoTSort and ByteTrack demonstrated a higher MOTA (Multiple Object Tracking Accuracy) score, which, as emphasized by Leal-Taixé (2017), is a critical measurement aligning closely with human visual perceptions of tracking accuracy.

Based on slightly lower ID switches compared to BoTSort, ByteTrack emerges as the recommended choice for a tracking algorithm in this context, highlighting its effectiveness in accurately tracking hornets in real-world scenarios. This research contributes to advancing the

understanding of tracking technologies within ecological and agricultural contexts, enabling more informed decisions for pest management and biodiversity conservation efforts.

Limitations and Future Work

This study's reliance on a single test video due to the challenges of generating ground truth data for multiple videos underscores a potential source of sampling error. To enhance the robustness of our findings, future research should incorporate data from multiple test videos to provide a more comprehensive evaluation of tracking algorithm performance for Hornet tracking.

Moreover, improving the quality and diversity of video datasets used for training and testing is crucial for optimizing tracking performance. High-resolution videos with detailed imagery are essential for accurately detecting and tracking small, fast-moving objects like flying hornets. Future studies should prioritize the inclusion of such datasets to refine the model's capabilities.

GitHub Code:

https://github.com/smewara/HornetsTracking

References

Aharon, N., Orfaig, R., & Bobrovsky, B. Z. (2022). BoT-SORT: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651.

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 3464-3468). IEEE.

Du, K., & Bobkov, A. (2023). An Overview of Object Detection and Tracking Algorithms. Engineering Proceedings, 33(1), p. 22.

Leal-Taixé, L., Milan, A., Schindler, K., Cremers, D., Reid, I., & Roth, S. (2017). Tracking the trackers: An analysis of the state of the art in multiple object tracking. arXiv preprint arXiv:1704.02781.

Lioy, S., Laurino, D., Maggiora, R., Milanesio, D., Saccani, M., Mazzoglio, P. J., Manino, A., & Porporato, M. (2021). Tracking the invasive hornet Vespa velutina in complex environments by means of a harmonic radar. Scientific Reports, 11(1), p. 12143.

Monceau, K., Bonnard, O., & Thiéry, D. (2014). Vespa velutina: A new invasive predator of honeybees in Europe. Journal of Pest Science, 87(1), pp. 1-16.

Sultana, F., Sufian, A., & Dutta, P. (2020). A review of object detection models based on convolutional neural network. Intelligent computing: Image processing based applications, pp. 1-16.

Ueno, T. (2014). Establishment of the invasive hornet Vespa velutina (Hymenoptera: Vespidae) in Japan. Int J Chem Environ Biol Sci, 2(4), pp. 220-222.

Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In 2017 IEEE International Conference on Image Processing (ICIP) (pp. 3645-3649). IEEE.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022, October). ByteTrack: Multi-object tracking by associating every detection box. In European Conference on Computer Vision (pp. 1-21). Cham: Springer Nature Switzerland.