informatics

MDPI

Article

The Relevance of Compound Events in Bee Traffic Monitoring

Andrea Nieves-Rivera, Marie Lluberes-Contreras * and Rémi Mégret *

Department of Computer Science, University of Puerto Rico at Río Piedras, San Juan, PR 00925, USA; andrea.nieves7@upr.edu

* Correspondence: marie.lluberes@upr.edu (M.L.-C.); remi.megret@upr.edu (R.M.)

Abstract

Bees are essential pollinators for agricultural systems, making accurate, automated monitoring of their behavior critical for assessing colony health and ecosystem stability. Recent advances in computer vision and artificial intelligence have enabled large-scale bee traffic monitoring at hive entrances; however, most existing event classification methods focus exclusively on simple entrance and exit events. This simplification overlooks compound movements—such as U-turns and guarding behaviors—that represent a substantial portion of bee activity and can lead to inaccurate trajectory reconstruction and misleading behavioral interpretations. In this work, we systematically analyze existing event classification strategies used in automatic bee traffic monitoring, evaluating their performance on both simple and compound movements. We then propose extended classification methods that explicitly model compound events by incorporating bidirectional movement patterns derived from positional and angular cues. Using a manually annotated dataset of computer-vision-based hive entrance recordings, we compare threshold-based, displacement-based, and angle-based approaches under simple and mixed-event conditions. Our results demonstrate that compound events account for over one-third of all detected movements and that classification methods explicitly designed to handle bidirectional behavior substantially outperform traditional approaches in both accuracy and robustness. In particular, threshold-based bidirectional classification achieves near-perfect performance when full trajectories are available, while displacement-based methods provide a reliable alternative under partial observations. These findings highlight the importance of modeling compound behaviors in automated bee monitoring systems and contribute to more accurate flight reconstruction, behavioral analysis, and AI-driven decision support for precision agriculture and pollinator management.

Keywords: bee traffic monitoring; compound event detection; computer vision; artificial intelligence; precision agriculture; pollinator behavior; trajectory reconstruction; hive entrance analysis

Check for updates

Academic Editors: Sarawut Ninsawat and Jaturong Som-ard

Received: 1 March 2026
Revised: 11 April 2026
Accepted: 16 April 2026
Published: 23 April 2026

1. Introduction

Bees are among the most important pollinators in agricultural ecosystems, playing a critical role in crop productivity and food security [1]. As a result, there is growing interest in automated, data-driven methods for monitoring bee activity [2–4] as indicators of colony health, environmental stressors, and ecosystem stability. In recent years, artificial intelligence and computer vision techniques have enabled continuous, non-invasive monitoring of bee traffic at hive entrances, producing large-scale spatiotemporal datasets suitable for computational analysis.

A central task in automated bee traffic monitoring is the classification of individual movements into meaningful behavioral events, most commonly entrances and exits [5].

 https://doi.org/10.3390/informatics13050065

These classified events serve as the foundation for higher-level analyses such as flight reconstruction [6], foraging activity estimation [7], and behavioral modeling [8]. Consequently, inaccuracies at the event classification stage can propagate through downstream analyses, leading to misleading interpretations of bee behavior and activity patterns.

Existing event classification methods are primarily designed to detect simple, unidirectional movements. However, observations at hive entrances reveal that a non-trivial proportion of bee activity consists of compound movements, such as U-turns [9], or guarding-related back-and-forth behavior [10]. When these compound events are forced into simple entrance or exit categories, trajectory reconstruction becomes ambiguous and behavioral signals are distorted.

Despite advances in computer-vision-based tracking, there is currently no standardized or widely adopted framework for modeling compound events in bee traffic data. Most implementations either misclassify these movements or discard them as noise. This gap highlights a broader informatics challenge: designing event classification algorithms that remain robust under complex, real-world behavioral dynamics while integrating seamlessly into automated data processing pipelines.

In this paper, we systematically evaluate existing event classification approaches used in automated bee traffic monitoring and analyze their limitations in the presence of compound movements. We then propose extended classification methods that explicitly model bidirectional behavior based on multiple approaches. By validating these methods against manually annotated ground truth data, we demonstrate that accounting for compound events substantially improves classification accuracy and supports more reliable AI-driven analysis of pollinator behavior in agricultural monitoring systems.

This paper is organized as follows. Section 2 reviews and organizes related work. Section 3 presents the proposed extensions and the methodology used to evaluate the diverse set of approaches. Section 4 presents the experimental results and the insights obtained from the detailed comparison of methods.

2. Related Works

Various methods have been proposed for tracking individual bees, ranging from Radio-Frequency IDentification (RFID) tags to fiducial and color-based tags. RFID initially replaced manual tracking due to its automation, allowing researchers to monitor a large number of individuals simultaneously [11] as each RFID tag can provide a unique ID when prompted by RF sensors, usually located in a controlled passage. RFID tagging has been used to observe differences between healthy bees and bees exposed to pesticides [12] as well as to analyze elitism in the foraging behavior among subcastes [13].

As RFID-based tracking became more widespread, a dedicated software tool, Track-a-Forager, was introduced in 2016 to facilitate data extraction and event classification from automatic detections [14]. This software extracts entrances and exits from RFID sensor data and reconstructs flight activity. It became a de facto standard for multiple works using RFID tags [15–18].

In recent years, fiducial tags combined with computer vision video analysis have emerged as an alternative for bee traffic monitoring, partly due to the cost of RFID tags [19]. This includes QR tags, AprilTag style fiducial markers [9,20,21], and BEEtag [22]. Computer vision is used to extract position, angle, pose and other features associated with each detection [23]. These types of automatic observation have been used for data extraction [24]. Some recent approaches also color tags [25–27].

Compared to RFID, computer vision approaches have higher computational requirements to perform detection, tracking and identification, but provide additional information, such as a more complete trajectory of bees at the entrance, and the presence of pollen [9] or

https://doi.org/10.3390/informatics13050065

parasites such as mites [28]. Additionally, this modality is increasingly accessible due to improvements in video capture and processing technology. For this reason, although some RFID-inspired approaches are presented, only approaches relevant to video analysis are discussed in the remainder of this paper.

In this section, we organize our analysis around the concept of events, defined as short intervals of time during which a bee is detected or visible by monitoring sensors, and which constitute the atomic unit of analysis that can be labeled with a specific behavior. In the case of RFID, this corresponds to a set of raw detections close in time, capturing the presence or the bee passing in front of sensors; for video this corresponds to a short-term trajectory before the bee leaves the field of view. We decompose the analysis into multiple steps for traffic monitoring: first, raw sensor detections, where individual RFID signals are collected or detections are obtained from individual video frames (Section 2.1); second, event detection, which generates short-term events by grouping raw detections (Section 2.2); and third, event classification which assigns a behavioral label to each event (Section 2.3).

2.1. Raw Sensor Detection

RFID detections rely on a bee passing through the detection range of a sensor. For event classification, two sensors are used to predict the trajectory of the bee based on the order in which they are triggered. For computer vision, two primary methods have been used: simple tag detection, and tag detection combined with pose estimation. Tag detection consists of reading the tag attached on the bee, which uniquely identifies each individual [29]. More advanced methods detect the positions of specific body parts [3,9,23].

2.2. Event Detection

Two primary approaches exist to track multiple detections across time and assign them to a single event: time clustering and computer-vision-based tracking methods.

2.2.1. Time Clustering

The time clustering approach assumes that events are short-lived and usually consists of groups of raw detections close in time, separated by longer periods during which the bee is out of view or away from the sensor, as is characteristic for bees passing in front of the sensor on their way in and out to forage outside of the hive. The name clustering and earliest formally defined implementation of this approach appears in Track-a-Forager [14] and has since been adopted by BeeCam-AprilTag [29,30]. This approach groups consecutive detections from the same individual based on the time interval between them. A predefined or user-specified threshold (in seconds) is used to verify whether two detections belong to the same event: if the time difference between consecutive detections does not exceed the threshold, they are considered part of the same event; otherwise, a new event is initiated. This method requires only the timestamp of each detection along with the individual bee's identifier. The resulting clusters represent continuous movements which can then be used to reconstruct trajectories.

2.2.2. Computer Vision Tracking

An alternative approach to event definition relies on computer vision tracking across consecutive video frames [5,24,31–33]. An event continues as long as the bee's position can be reliably tracked using both spatial and temporal continuity. The event ends once the tracking is lost. This approach requires access to video data or individual frames. An individual identifier for the bees is required to maintain continuity across frames.

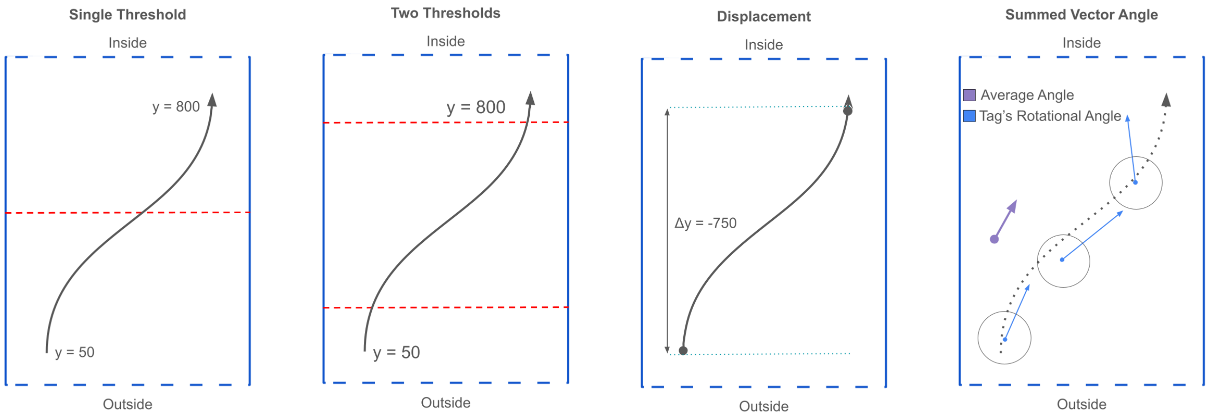
2.3. Event Classification Methods

We group event classification methods into three categories based on the type of information they rely on: threshold-based, displacement-based, and angle-based. Most existing methods only consider two classes of interest: entering or exiting. Some approaches include a reject class (e.g., unknown, noise or undetermined), with a few works defining more explicit classes for some edge cases, such as flying-by [5], walking [9], lateral [33]. These implementations and their corresponding outputs are summarized in Table 1 and discussed in more detail below.

Table 1. Classification of existing implementations.

Implementation	Heuristic	Classes	Details
Chen et al. (2012) [24]	Displacement-based	entering, exiting	Uses initial and final position of an event.
Tu et al. (2016) [31]	Threshold-based	entering, exiting	Uses two areas with a single threshold, flying-in and flying-out.
Track-a-Forager (2016) [14]	Threshold-based	entering, exiting	Uses RFID sensors.
Magnier et al. (2018) [5]	Threshold-based	entrance, departure, flying-by	Uses two areas with a single threshold, inside-zone and outside-zone.
Mukherjee & Kulyukin (2020) [33]	Angle-based and displacement-based	outgoing, incoming, lateral	The displacement angle of each movement is obtained.
Rodriguez et al. (2022) [9]	Threshold-based	entering, leaving, walking, noise	Uses two thresholds and three zones: inside, outside, and ramp.
Le et al. (2024) [32]	Displacement-based	flying-in, flying-out, undetermined	Offline approach uses initial and final position of an event.
Lei et al. (2024) [34]	Threshold-based	incoming, outgoing, error	Verifies if a bee passes a threshold.
Lei et al. (2024) (Box method) [34]	Threshold-based	incoming, outgoing, error	Propose custom box shaped thresholds for the case the ramp is open on the sides.
BeeCam–AprilTag (2025) [29]	Angle-based	exit, enter, unknown	Obtains a directional vector from the rotational angle of the bee.

In the following, we adopt the convention that the hive entrance is oriented such that upward movement (decreasing y-coordinate) represents entrance into the hive, and downward movement (increasing y-coordinate) represents exit from the hive (see Figure 1). The sides of the ramp are assumed to be closed, as in most existing works, with an alternative method to account for open sides [34].



(**a**) (**b**) (**c**) (**d**)

Figure 1. Different approaches illustrated, with a gray arrow representing the bee's trajectory and red dashed lines representing thresholds. Dashed blue lines mark entrance and exit to the hive. (**a**) Single threshold Tu et al. [31] and Magnier et al. [5] referred to as `THRS_SINGLE`. (**b**) Two thresholds Rodriguez et al. [9] also referred to as `THRS_DOUBLE`. (**c**) Initial and final position used for displacement in Chen et al. [24] and Mukherjee and Kulyukin [33], also referred to as `DISP_FULL`. (**d**) Summed vector direction from angles utilized in BeeCam–AprilTag [29,30], also referred to as `ANGLE_SUM`.

2.3.1. Threshold-Based Classification

Threshold-based methods classify events based on whether the bee's movement crosses one or more predefined thresholds. If the bee traverses the threshold(s), the event is classified accordingly; otherwise it is assigned an unknown or ambiguous label. This approach is employed by Tu et al. [31] with a single threshold dividing the entrance into flying-in and flying-out, and by Magnier et al. [5] with a single threshold dividing the entrance into inside zone and outside zone, as illustrated in Figure 1a. Lei et al. [34] use a similar approach with the threshold placed at the beginning of the entrance but do not track individual bees, instead focusing on counting entrances and exits across all bees. A second implementation, referred to as the box method, utilizes two thresholds in cases where the ramp has no side walls, allowing events to be classified when a bee enters from the side rather than from the top or bottom. For such single threshold approaches, an entering event is therefore defined as starting in the outside zone and ending in the inside zone, and an exiting event is the opposite. Starting and ending in the same zone is typically discarded. In the rest of this paper, we will refer to the approach with a single threshold dividing the entrance ramp in two as `THRS_SINGLE`.

Rodriguez et al. [9] use two thresholds that partition the entrance into inside, ramp and outside regions, as illustrated in Figure 1b. Their approach classifies trajectories that begin and end in the ramp region as walking, and ignore cases where the trajectory both begins and ends in either the outside or inside region by labeling them as noise. The implementation is referred to as `THRS_DOUBLE`, which we extend in Section 3.4.1 to `THRS_BIDIRECTIONAL` by defining how the noise class can actually be further divided into different compound cases.

2.3.2. Displacement-Based Classification

Displacement-based classification operates on the positional change between the initial and final detection of a bee relative to the camera [24,32]. The difference in y-coordinates between two detections associated with an event is used to infer the bee's trajectory. Both implementations rely on the first and last detection in an event to classify the displacement, and are collectively referred to as `DISP_FULL`. This approach requires at least two positional detections to determine movement direction, as illustrated in Figure 1c, and is not applicable in cases where only a single detection is available. We propose extensions on how to compute the displacement to take into account compound events in Section 3.4.2.

2.3.3. Angle-Based Classification

Angle-based classification relies on a directional vector derived from detection data. This vector can be obtained from the angle of a fiducial tag, as implemented in BeeCam-AprilTag [29,30], or from a displacement vector [33]. In both cases, the resulting angle is compared with user-specified thresholds to classify each event as entrance or exit.

The tag-angle implementation can consistently derive a direction from detections even when only a single positional coordinate is available, while maintaining a low computational cost. However, it is sensitive to the erratic behavior of bees. BeeCam-AprilTag computes a summed vector direction across all angles within an event (hereafter referred to as `ANGLE_SUM`), as illustrated in Figure 1d, smooths out the individual detection angles to obtain the general trend from each track.

In contrast, the displacement-angle approach requires at least two positional detections to obtain a direction [33]. However, this implementation considers each displacement between frames individually, which introduces the risk of counting the same track multiple times.

We propose an extension to the BeeCam-AprilTag sum angle approach [29,30] to take into account the specific track shape of compound events in Section 3.4.3.

3. Methods

This section describes the proposed extensions to event classification, including two designed to model compound movements, as well as the experimental design used to assess their performance. The proposed approaches are compared to the existing methods described in Section 2.3. All methods are implemented by the authors and operate on the same event representation, corresponding to a sequence of detections associated with a single bee at the hive entrance. An event is represented by a temporally ordered list of detections, each containing positional and/or angular information.

To provide a controlled and consistent comparison using identical data, preprocessing, and evaluation conditions, none of the algorithms are adopted directly from existing software tools; instead, all are implemented or re-implemented within a unified framework and evaluated against the same manually annotated ground truth.

Our objective is twofold: (i) to evaluate the accuracy of existing event classification heuristics under both simple and compound movement conditions, and (ii) to propose algorithmic extensions that explicitly model bidirectional behavior such as U-turns within a single event. To this end, we implement and compare threshold-based, displacement-based, and angle-based methods, along with bidirectional variants that combine initial and final movement cues. These new classes are entering-exiting, and exiting-entering, which represent movements in which a bee took a U-turn mid-path, as illustrated in Figure 2.

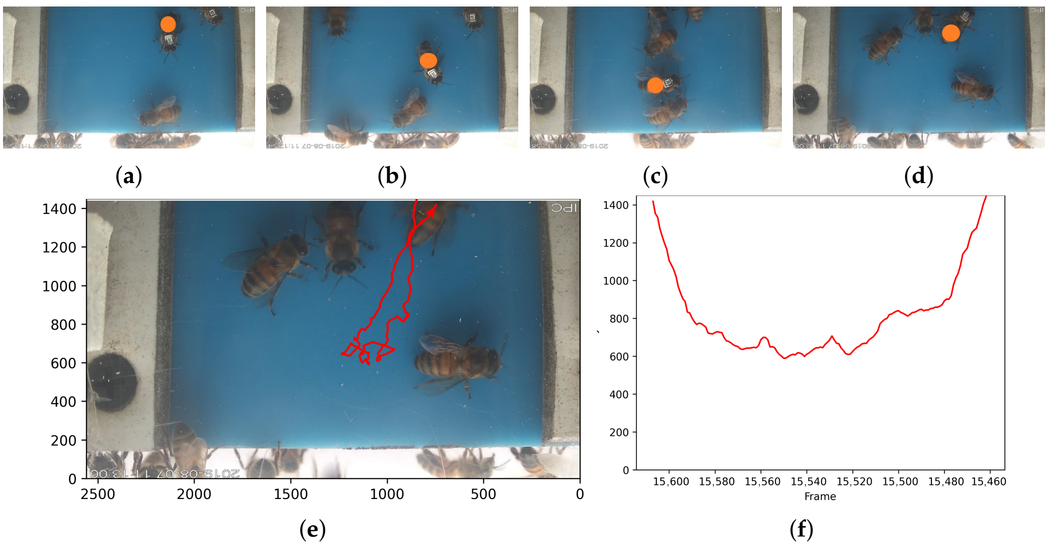


Figure 2. Example of U-turn on the ramp (exit-entrance): (**a**–**d**) Bee movement across frames. (**e**) Trajectory of the bee overlayed on the last frame of the trajectory. (**f**) Bee movement on y-position over frames.

3.1. Rationale of the Proposed Approaches

Mapping the y-positions of detections across frames reveals the occurrence of these movements types, such as in Figure 3a, which represents an outward movement followed by an inward movement. When assembling flights, these subsequent entrances are typically ignored due to the absence of an exit between them; however, retaining them can provide additional information about bee behavior, such as sentry activity or social interaction. Figure 3b represents the opposite pattern: an inward movement that is often preceded by another outward movement. When assembling flights with only exits and entrances, only the last exit followed by an entrance is recorded. As a result, events such as those in

Figure 3b are ignored, despite the fact that they can be considered a termination of a flight. These two types of movements are referred to as compound events. Existing methods are primarily designed for movements such as those shown in Figure 3c, which represent clear exit movements and are referred to as simple events.

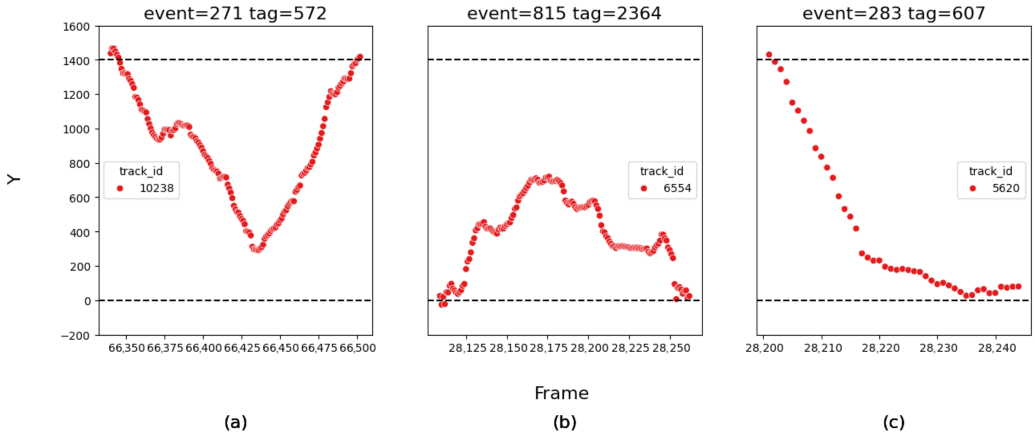


Figure 3. Different movements at the entrance of the hive: (**a**) Graph for event 271 showing an outward movement followed by an inward movement without an exit in between. (**b**) Graph for event 815 showing an inward movement followed by an outward movement without an entrance in between. (**c**) Clear event of an exiting motion in event 283.

3.2. Event Representation and Assumptions

Each event $E = \{d_1, d_2, \ldots, d_N\}$ consists of N detections corresponding to a continuous interaction of a bee with the hive entrance. Each detection include: a frame index, a two-dimensional position relative to the camera, an angle derived from the bee pose (or marker orientation), and an identifier obtained from the fiducial marker.

All methods assume that events have already been defined through computer-vision-based tracking and time clustering, as described in Section 2.2.

3.3. Evaluated Methods

In total, eleven classification methods were evaluated, grouped in the 3 categories described in Section 2.3. The seven methods introduced in this work are marked with (new) at the end of their name and explained in more detail in Section 3.4.

1. Threshold approaches
    (a) `THRS_SINGLE` (used in [5,31,34]): The y range was divided into two areas with a single threshold, inside and outside. The threshold was placed in the middle of the y range, at $y = 770$.
    (b) `THRS_DOUBLE` (used in [9]): The y range was divided into three areas with two thresholds as provided by the data, at $y = 450$ and $y = 600$. The thresholds were selected such that a full sized bee fits between each border of the image (entrance and exit sides) and the corresponding threshold. This ensures that detection can rely on fully visible bee before crossing the threshold.
    (c) `THRS_BIDIRECTIONAL` (new): Considers movements in which the bee starts and ends in the same threshold-divided area as compound movements, see Section 3.4.1. Uses the same thresholds as `THRS_DOUBLE`.
2. Displacement approaches
    (a) `DISP_FULL` (used in [32]): Computes the displacement based on the full track (difference between the initial and final y position of an event), and classifies based on whether the difference is negative or positive.

   (b) `DISP_START` (new): Computes the displacement based on the start of the track only, see Section 3.4.2.
   (c) `DISP_END` (new): Computes the displacement based on the end of the track only, see Section 3.4.2.
   (d) `DISP_BIDIRECTIONAL` (new): Combination of `DISP_START` and `DISP_END` to account for compound events, see Section 3.4.2.

3. Angle approaches
   (a) `ANGLE_SUM` (used in [29]): Summed vector direction of the bee angle implemented by BeeCam-AprilTag. This method designates entrance as an angle of $90° \pm 60°$ and exit as $270° \pm 60°$.
   (b) `ANGLE_START` (new): Uses the start of the track to compute the summed vector direction, then used in the same way as for `ANGLE_SUM`, see Section 3.4.3.
   (c) `ANGLE_END` (new): Uses the end of the track to compute the summed vector direction, then used in the same way as for `ANGLE_SUM`, see Section 3.4.3.
   (d) `ANGLE_BIDIRECTIONAL` (new): Combination of `ANGLE_START` and `ANGLE_END` to account for compound events, see Section 3.4.3.

3.4. Proposed Methods

The seven proposed methods are designed to operate on event-level data and extend existing approaches by adapting similar heuristics but in a way that enables proper classification in the case of a compound event of type entrance-exit or exit-entrance.

3.4.1. Threshold-Based

`THRS_BIDIRECTIONAL`: The initial and final location of the bee are considered in relation to two dividing thresholds, similarly to `THRS_DOUBLE`, which splits the ramp into three regions: inside, ramp and outside. When both the initial and final regions are the same, they are classified as compound events: inside-inside corresponds to an exiting-entering event, and outside-outside corresponds to an entering-exiting event. Trajectories starting or ending in the ramp region are classified as unknown, to account for the premature interruption of tracking. This is presented in Algorithm 1.

Algorithm 1 Compound Event Classification: THRS_BIDIRECTIONAL

```
Input: Initial and final area of bee
Output: Compound event C
if init = end then
   if init = 'entrance' then
      return 'exiting-entering'
   else if init = 'exit' then
      return 'entering-exiting'
   else
      return 'unknown'
   end if
else
   if end = 'exit' then
      return 'exiting'
   else if end = 'entrance' then
      return 'entering'
   else
      return 'unknown'
   end if
end if
```

3.4.2. Displacement-Based

Displacement-based methods infer movement direction from changes in a bee's position along the vertical axis of the image, which corresponds to inward or outward motion relative to the hive, using 100 pixels as the distance indicator for direction; empirically chosen. The existing `DISP_FULL` heuristic cannot identify a U-turn because it considers only the relative position of the start and end points. For this reason, we define displacement criteria specific to either the start or end reference point to evaluate how far the trajectory deviates from that reference.

- `DISP_START`: This approach classifies a bee's trajectory as entering or exiting based on the change in its y-coordinate within a single event. Specifically, it compares the y-position of the first detection with an intermediate detection. This intermediate detection starts from the second detection and iterates forwards until a displacement exceeding a predefined threshold is found. As a result, at least two distinct y-positions are required. The classification results in an unknown label when the initial and final y-positions are identical. This is described in Algorithm 2.
- `DISP_END`: A modified variant applies the same procedure in reverse to determine the final direction of the bee's movement. Following the same method but iterating backward, this approach is described in Algorithm 3.
- `DISP_BIDIRECTIONAL`: The approach combines previous `DISP_START` and `DISP_END` variants to handle u-turns following the approach described in Section 3.4.4.

Algorithm 2 Event Classification: DISP_START

```
Input: List of all y positions in detections of an event E, user-defined distance threshold
Output: Classified Event C
N ← len(E)
if N < 2 then
  return unknown
end if
start ← E_0
for k = 1 to N do
  end ← E_k
  if |start − end| ≥ threshold then
    break
  end if
end for
if start > end then
  return exiting
else if start < end then
  return entering
else
  return unknown
end if
```

3.4.3. Angle-Based

Angle-based methods designate entrance as an angle of $90^\circ \pm$ angle_threshold and exit as $270^\circ \pm$ angle_threshold. In [29], the parameter angle_threshold $= 60^\circ$ was chosen. We will use the same value. To generalize angle-based methods, we infer the event class from the orientation of the bee using the average angle vectors over the first and the second half of the trajectory, rather than using an average across the entire trajectory.

- `ANGLE_START`: The angle of summed angle vector over the first half of the trajectory is used to define the class. This angle is compared against predefined thresholds to classify the movement as entering, exiting or unknown. The angle is first normalized to a range between $0°$ and $360°$, with $0°$ corresponding to the rightmost direction, and then classified. This method is described in Algorithm 4.

- `ANGLE_END`: A complementary variant using only the angle of the summed vector over the second half of the trajectory and is described in Algorithm 5.
- `ANGLE_BIDIRECTIONAL`: The approach combines previous `ANGLE_START` and `ANGLE_END` variants to handle u-turns following the approach described in Section 3.4.4.

Algorithm 3 Event Classification: DISP_END

Input: List of all y positions in detections of an event E, user-defined distance *threshold*
Output: Classified Event C
if $N < 2$ **then**
  return *unknown*
end if
$end \leftarrow E_N$
for $k = 1$ **to** $N - 1$ **do**
  $init \leftarrow E_{N-k}$
  if $|init - end| \geq threshold$ **then**
    break
  end if
end for
if $end < init$ **then**
  return *exiting*
else if $init < end$ **then**
  return *entering*
else
  return *unknown*
end if

Algorithm 4 Event Classification: ANGLE_START

Input: Summed vector angle over the first half of the trajectory in an event *angle*, user-defined thresholds $exitmax, exitmin, entermax, entermin$
Output: Classified Event C
$degrees \leftarrow angle \bmod 360$
if $exitmin \leq degrees \leq exitmax$ **then**
  return *exiting*
else if $entermin \leq degrees \leq entermax$ **then**
  return *entering*
else
  return *unknown*
end if

Algorithm 5 Event Classification: ANGLE_END

Input: Summed vector angle over the second half of the trajectory in an event *angle*, user-defined thresholds $exitmax, exitmin, entermax, entermin$
Output: Classified Event C
$degrees \leftarrow angle \bmod 360$
if $exitmin \leq degrees \leq exitmax$ **then**
  return *exiting*
else if $entermin \leq degrees \leq entermax$ **then**
  return *entering*
else
  return *unknown*
end if

3.4.4. Combination of `START` and `END` Estimates into `BIDIRECTIONAL` Estimates

By combining class information from both the initial and final portions of a trajectory, it is possible to classify events as entering-exiting or exiting-entering. When the classes inferred from the initial (`DISP_START`, `ANGLE_START`) and final parts (`DISP_END`, `ANGLE_END`) of a trajectory differ, they are combined into new compound event classes entering–exiting or exiting–entering. When they are the same, the event is labeled as simple class entering or exiting. The respective bidirectional algorithms are referred to as `DISP_BIDIRECTIONAL` and `ANGLE_BIDIRECTIONAL`.

3.5. Experimental Evaluation Methodology

3.5.1. Video and Track Datasets

To evaluate existing classification approaches and compare them against the proposed methods, we used video data consisting of hour-long recordings at a resolution of 2560 × 1440 pixels and 20 frames per second, captured at a hive entrance in Gurabo, Puerto Rico. Figure 2 shows example of frames from such video. In this setup, a transparent acrylic sheet was present on top of the ramp, to ensure that all bees entering or exiting would be forced to pass through the restricted entrance in the 2D plane of focus of the camera. The camera is a 4 Mpixels IP camera (GESS Technology, model GESSXDD-IP740QMVR-W3 333, San Juan, PR, USA) recording H.264 video at 8 Mbps.

Videos were processed using YOLO11x pose detection model [35] trained to detect the following honey bees body keypoints: head, neck, thorax, waist and tip of the abdomen. AprilTag qrcodes were detected in the thorax region of each bee using code adapted from the apriltag library [20]. The 25h5 family of 5 × 5 apriltags was used in these videos. The bee orientation angle was defined by the angle between the abdomen tip and the head. Although it is a more computationally expensive approach compared to the use of tag angle in [29], this offered several advantages: this angle was available even when the tag was not detected (bees rotated on the side, or blurry tag) as pose was detected much more consistently, and it represented the actual body orientation. Using the tag angle as a proxy was biased in some cases due to the practical difficulty of gluing a perfectly aligned tag on the thorax.

All bee detections were tracked using the Hungarian algorithm as in [9] to capture complete bee movements, and the resulting tracks were filtered to retain only bees identified with an AprilTag on their thorax. Keeping only the tagged bees for this study was motivated by the human validation process to ensure the quality of the tracks used for evaluation. Indeed, the visual inspection of the tracks is facilitated by the presence of the tag in the typical ramp crowded by multiple bees. It should not be considered a limitation of the event classification itself, as tag presence was not required for the generation of the tracks themselves. The purpose of the tracking in this work was to provide clean validated real trajectories for event analysis, and we refer the reader to existing literature such as [9,34] for evaluation of the detection and tracking approaches themselves.

Each detection in the raw dataset includes the video name, frame number, track identifier, spatial position, bee angle, and a unique bee identifier derived from the AprilTag. An example of video frames and trajectory is shown in Figure 2.

3.5.2. Event Datasets

A dataset of manually annotated events was created from the data extracted from 12 consecutive one hour-long 20 fps videos between 11:00 and 17:00 on 7 and 8 August 2019, for detailed evaluation of the classification algorithms.

The processing pipeline described in Section 3.5.1 produced 215,035 raw detections corresponding to detections of the same population of bees over time. From these detections, tracking generated 1690 raw tracks with an average of 127.2 frames (6 s) and a median of 62 frames (3 s).

To address track fragmentation caused by temporary occlusion or bees leaving the field of view for very short time, a time-clustering window of 5 s was applied to merge tracks corresponding to the same individual. After clustering, the dataset was reduced to 1421 distinct events as the process merged multiple consecutive tracks into single aggregated events. These clustered events have an average of total length from start to end of 151 frames (8 s) and a median of 72 frames (4 s). 5 s were chosen though trial and error after initially using 20 s as in [14], but this window resulted in many tracks with misclassified or artificially created events due to several wrongly concatenated tracks. A window of 60 s, as used in [29], resulted on several disjointed events pasted together producing wrongly classified compound events.

These events were visualized and manually classified to establish ground truth labels for the 4 classes of interest: entering, exiting, entering-exiting, exiting-entering. Manual classification was performed primarily by visualizing the trajectory of the y coordinate of the central point of the bee (waist keypoint) with respect to time for each event. When the start and end of the trajectory were clearly initiated at the top or bottom border of the frame, the event was classified accordingly into simple events (entering, exiting), or U-turns (entering-exiting or exiting-entering). When such patterns were ambiguous (interrupted track, discontinuity, start or end away from the top or bottom borders, absence of consistent tag detection), classifications were confirmed by inspecting the corresponding videos directly. Tracks not corresponding to a properly tagged bee, or due to tracking errors were excluded to focus on the problem of behavior classification from clean data, resulting in 1194 events. These manually assigned labels were treated as ground truth for all experimental evaluations.

A larger, non-annotated dataset consisting of 289 h-long videos was also created. It contains 3,567,840 raw detections, resulting in 22,820 tracks following the same preprocessing pipeline as the smaller annotated dataset. This dataset was not manually annotated, but was used to analyze the prevalence of events over an extended period of 23 days.

3.5.3. Experimental Design

The classification performance of the algorithms was evaluated in three experiments, focusing on precision, recall, and accuracy. Metrics were computed using `scikit-learn`'s classification report.

In the first experiment, simple algorithms were tested on both simple and compound events, with compound events grouped into a reject class labeled as unknown. This allowed evaluation of each algorithm's ability to correctly classify simple events while rejecting non-simple ones.

In the second experiment, compound-capable algorithms were evaluated in a multi-class setting to assess their ability to distinguish between simple and compound event types.

In the third experiment, the prevalence of compound behavior at scale was evaluated using the full 289-h dataset. All 22,820 events were classified using compound-capable algorithms to estimate the proportion of entering–exiting and exiting–entering movements.

4. Results

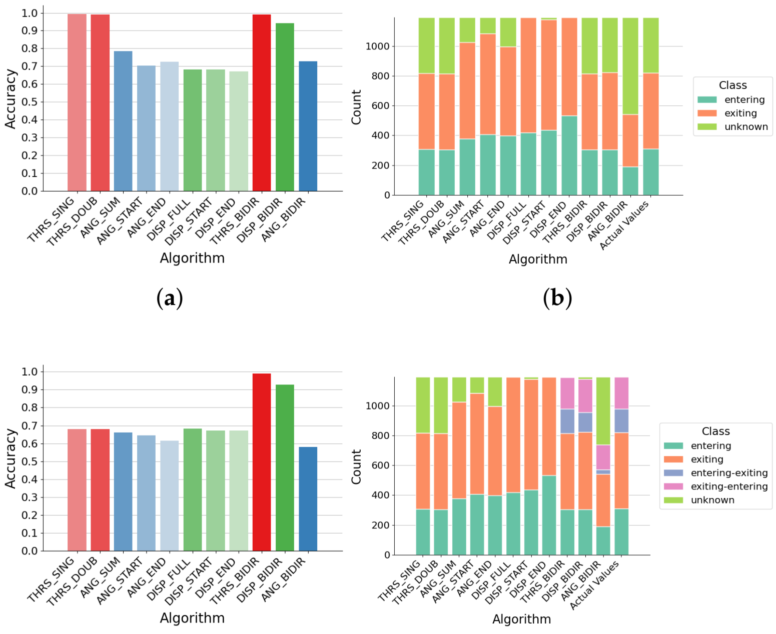
This section presents the results of the experimental evaluation in four parts. We first provide a general overview of classification performance across all algorithms under simple and compound conditions (Section 4.1). We then analyze compound-event classification in detail (Section 4.2), followed by a focused evaluation on simple events only (Section 4.3). Finally, we examine the prevalence of compound events in the full dataset (Section 4.4).

4.1. General Classification Performance

To evaluate the general behavior of all algorithms, we first computed classification accuracy under conditions most favorable to existing approaches by merging the compound events class into a reject class labeled unknown. This analysis assesses each algorithm's ability to recognize simple events while properly rejecting non-simple event. Accuracy is shown in Figure 4a.

Algorithms that always assign a simple classification—namely the simple displacement-based and angle-based methods—exhibit low accuracy under this setting, as they rarely assign the unknown label. Figure 4b presents the distribution of predicted classes for each algorithm, enabling comparison with the true event distribution. The proposed displacement-based methods tend to map non-simple events to exiting classifications. In contrast, the proposed bidirectional methods achieve substantially higher accuracy by explicitly modeling non-simple movements. Simple threshold-based methods also yield unknown classifications when events start and end in the same region, contributing to their relatively high accuracy in this setting. Although they are not designed to recognize specific compound events, they can correctly reject them as neither entering nor exiting.

We next evaluated performance when both simple and compound events were included as explicit classes. Accuracy and class distributions are shown in Figure 4c,d. Under this four-class setting, all simple classification algorithms fall below 70%, whereas the compound-capable algorithms `THRS_BIDIRECTIONAL` and `DISP_BIDIRECTIONAL` maintain accuracy above 90%. `ANGLE_BIDIRECTIONAL` falls below all other algorithms, with an increased drop placing it below 60%.



(**a**) (**b**)

(**c**) (**d**)

Figure 4. Metrics for three classes (entering, exiting, unknown) and four classes (entering, exiting, entering-exiting, exiting-entering). (**a**) Accuracy for three classes. (**b**) Distribution of predictions for three classes. (**c**) Accuracy for four classes. (**d**) Distribution of predictions for four classes.

In this setting, the unknown label refers to events that could not be confidently assigned to any of the four classes. `ANGLE_BIDIRECTIONAL` exhibits the highest proportion of unknown classifications, which is examined further in Section 4.2.

Overall, these results demonstrate that, due to the prevalence of compound events in the dataset, accuracy is significantly affected when they are not modeled. If detailed recognition of compound events is not required and they can be treated as a noise class, simple threshold-based approaches perform well in rejecting them, which is not the case for angle-based or displacement-based approaches.

4.2. Compound Events

To analyze compound-event behavior in greater detail, we computed precision and recall for all four classes. These results are shown in Figures 5a and 5b, respectively.

`DISP_BIDIRECTIONAL` achieves high precision across all classes, particularly for simple events, indicating reliable predictions. Recall remains strong for simple events but decreases for compound behaviors. Performance is asymmetric across compound classes, with entering–exiting events being less consistently detected. Misclassification patterns suggest that entering–exiting events are frequently reduced to simple classes, indicating that the method struggles to preserve bidirectional structure in compound trajectories.

The limitations of angle-based methods become more evident in the compound setting. When combining `ANGLE_START` and `ANGLE_END`, a substantial fraction of detections are classified as unknown, resulting in compound events that cannot be further interpreted. This behavior is reflected in Figure 4d, where a large proportion of `ANGLE_BIDIRECTIONAL` outputs fall into the unknown class. Its precision is highest for exiting events but it doesn't outperform the other algorithms, with substantially poorer performance across the remaining classes. The overall accuracy of this method is 0.579 indicating limited suitability when full trajectory information is available.

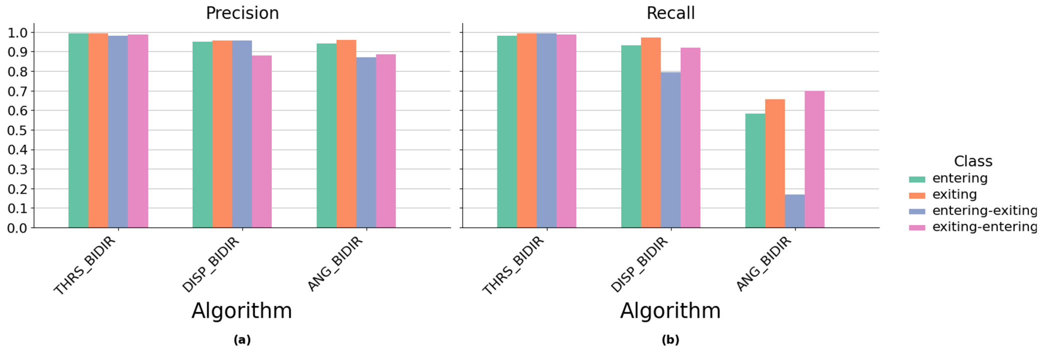


Figure 5. Metrics for compound events. (**a**) Precision for compound events. (**b**) Recall for compound events.

In contrast, `THRS_BIDIRECTIONAL` achieves very high precision and recall for compound events. Events in which a bee briefly appears and leaves the entrance region are effectively classified using initial and final positions. This method achieves near-perfect recall for both entering–exiting and exiting–entering events, along with near perfect classification of simple events in this dataset. Its primary limitation arises when tracks are interrupted in the middle of the ramp and tracking is lost. This type issues are currently reported as unknown, but could motivate future work for proper imputation of incomplete trajectories. Overall, these results indicate that `THRS_BIDIRECTIONAL` is the most effective algorithm when complete trajectory information is available. This method classifies as compound events 31.5% of the 1194 events in the manually annotated dataset and correctly classifies 370 of the 374 compound events in the set. Confusion matrices of the 3 bidirectional methods are shown in Appendix A.

4.3. Simple Events

To provide additional insights in the reasons for success or failure of the diverse algorithms, we evaluate their behavior on simple events by restricting the dataset to entering and exiting events only. Under this setting, all algorithms achieve very high precision, and all algorithms except ANGLE_START and ANGLE_END achieve high recall for both classes, as shown in Figure 6a,b.

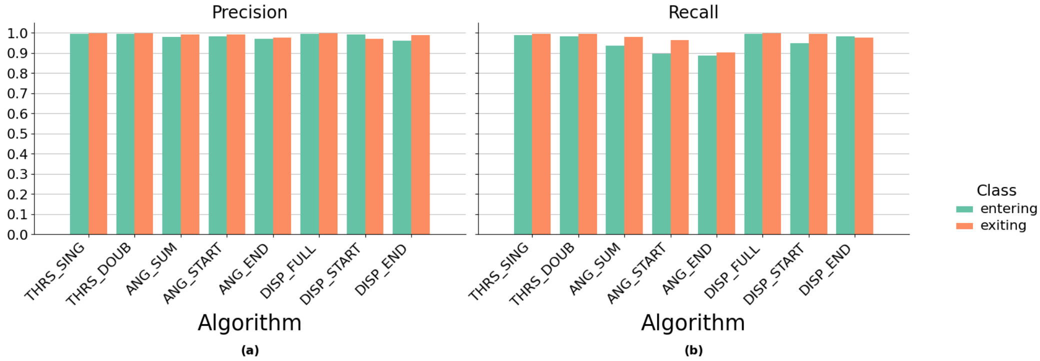


Figure 6. Metrics for simple events. (**a**) Precision for simple events. (**b**) Recall for simple events.

ANGLE_START exhibits a lower recall for entering events. In comparison, ANGLE_END shows reduced recall for both exiting and entering events. This behavior is likely due to the final rotational angle being more variable and noisy near the exit region, where a wider range of orientations does not strongly indicate direction. As a result, ANGLE_BIDIRECTIONAL fails to reliably classify both simple and compound events.

Among the newly proposed heuristics, DISP_START and DISP_END achieve high precision and recall for both entering and exiting events. A slight performance difference can be observed between the entrance and exit classes, likely influenced by erratic bee motion near the hive exit. Nevertheless, their performance is comparable to existing approaches, with overall accuracy values of 0.976. DISP_END achieves performance comparable to DISP_FULL, indicating that final displacement captures most of the discriminative information for simple-event classification. DISP_START also performs strongly but exhibits a slight reduction in recall for entering events, suggesting greater sensitivity to noise or ambiguity in initial trajectory segments. The combination of both components in DISP_FULL provides a more robust representation, although only marginal improvements over DISP_END are observed. While DISP_FULL performs best overall, the strong performance of DISP_END indicates that simple-event classification is primarily governed by endpoint information. This helps explain why displacement-based methods perform well in simple cases but require additional structure to handle compound events.

4.4. Large-Scale Compound Event Prevalence

Finally, to confirm the consistent prevalence of compound events, all 22,820 events extracted from the full dataset were classified using the compound-capable algorithms. This resulted in 8430 compound events detected by the best performing method THRS_BIDIRECTIONAL, accounting for 36.9% of all events. DISP_BIDIRECTIONAL and ANGLE_BIDIRECTIONAL's compound event classification percentages are 26.3% and 25.1%, respectively. We also compared to the number of events classified with a clustering window of 20 s, which account for 30.7% using THRS_BIDIRECTIONAL. As shown in Figure 7, a large proportion of compound events correspond to entering–exiting movements, indicating aborted or short-lived entering attempts that would be misclassified under simple event models and may introduce errors in downstream analyses. The mean percentage of unknown classifications, corresponding to tracks that end in the ramp area, across all days is 0.4%, while the median is 0.2%.

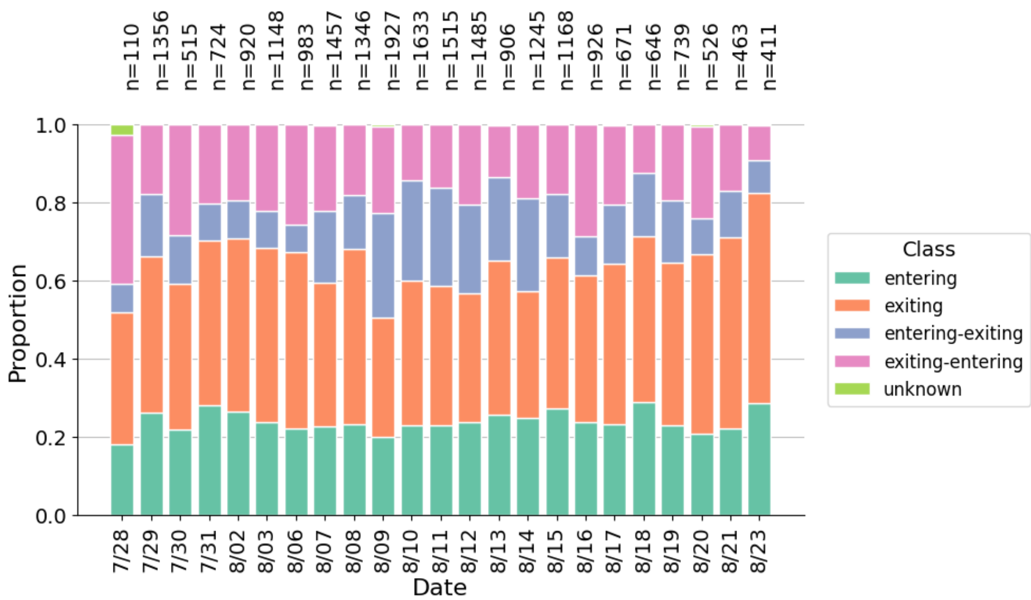


Figure 7. Class distribution for four classes on the full dataset using `THRS_BIDIRECTIONAL`. The figure illustrates the number of events classified in each category across the dataset for each day. The number over the bars represents the number of events recorded on that day for a total of 22,820.

4.5. Impact in Flight Assembly

To analyze the impact of compound movements in flight assembly, we compared the resulting flights in the dataset after using different algorithms for event classification. For this analysis, we selected the two best performing algorithms for compound movements and two existing approaches from different categories. The selected algorithms are: `THRS_BIDIRECTIONAL`, `DISP_BIDIRECTIONAL`, `DISP_FULL` and `ANGLE_SUM`. We define a flight F_i as an exit event for which the closest following event is an entry into the hive, with the time passed between these pair of events representing the flight itself. Formally, $F_i = (t_{s_i}, t_{e_i})$ where t_{s_i} is the start time and is associated with an exiting event, t_{e_i} is the end time of the flight and associated with an entering event, and $t_{e_i} = t_{s_i} + \delta, \delta \geq 120''$. The Table 2 shows the results of these methods compared to manually assembled flights. Out of 131 true flights based on the manually annotated ground truth, events classified using `THRS_BIDIRECTIONAL` correctly assembled 130 flights and `DISP_BIDIRECTIONAL` 117. Both methods show equal number of 3 false positive flights assembled, and 1 and 14 false negative, for 99% and 96% accuracy, respectively. These were the highest true positive and accuracy values out of all four methods. The highest number of false negatives was obtained by `ANGLE_SUM` while `DISP_FULL` shows the highest number of false positives. The results shown in Table 2 serve as preliminary evidence of the relevance of compound events in flight assembly.

Table 2. Flight assembly results for intervals of over 2 min under 5 h. Based on the manual ground-truth, the total of flight intervals is 131 (Positives) and the total of non-flight intervals is 318 (Negatives).

Algorithm	True Positive	False Positive	True Negative	False Negative
`DISP_FULL`	112	13	305	19
`ANGLE_SUM`	102	9	309	29
`DISP_BIDIRECTIONAL`	117	3	315	14
`THRS_BIDIRECTIONAL`	130	3	315	1

5. Conclusions

This study demonstrates that compound movements constitute a substantial and previously under-modeled component of bee traffic at hive entrances. As a consequence, ignoring these events during automated classification introduces the risk of systematic errors in foraging flight reconstruction and other downstream behavioral analyses that rely on determining whether an individual bee is inside or outside of the colony. Our results show that event classification algorithms explicitly designed to capture bidirectional movement patterns are significantly more robust than traditional methods that assume strictly unidirectional behavior.

Among the evaluated approaches, threshold-based bidirectional classification performs best when full trajectories are available, offering high accuracy for both simple and compound events. These findings emphasize the importance of incorporating behavioral complexity into the design of event classification algorithms rather than modeling only the primary classes of interest (in this case, entrance and exit events) and treating deviations as noise or ambiguity.

By extending existing heuristics to explicitly model compound events, we improved the interpretability and fidelity of automated monitoring systems without increasing computational complexity. The proposed methods can be readily integrated into computer-vision-based tracking workflows and adapted to other domains involving dense, bidirectional movement near boundaries.

Ultimately, incorporating compound event modeling into AI-driven bee monitoring systems enables more accurate behavioral analysis and supports data-informed decision making in precision agriculture and pollinator management. More broadly, this work underscores the role of robust event classification as a foundational component of intelligent environmental monitoring systems. Preliminary results show that the number of bee's flights missed due to incorrect event classification show improvement up to over 20% when flights are assembled using compound event classification methods.

This work used data with good temporal resolution for tracking (20 fps), as well as using the body detections to perform tracking when the tag was not detected, providing complete tracks that were validated manually for evaluation on a clean dataset. The displacement and angle based heuristics under-performed the simpler threshold based approach in that context, and they may mistake compound events for simple events even when the full track data is available.

In practice, edge devices deployed in the field may not have the computational budget for complete pose estimation, tracking and robust tag detection at high frame rate to obtain clean tracks, which is an extra challenge that was not considered in this work. As future work, this could be addressed by evaluating the impact of such incomplete information on event classification performance. Based on the findings from this paper, we believe this should explicitly take into account the presence of compound events, and may lead to hybrid approaches where incomplete tracks interrupted on the ramp could be imputed based on angle or displacement information.

Author Contributions: Conceptualization, R.M.; methodology, R.M. and M.L.-C.; software, A.N.-R.; validation, A.N.-R.; formal analysis, R.M. and M.L.-C.; investigation, A.N.-R., M.L.-C. and R.M.; resources, M.L.-C. and R.M.; data curation, A.N.-R.; writing—original draft preparation, A.N.-R. and M.L.-C.; writing—review and editing, A.N.-R., M.L.-C. and R.M.; visualization, A.N.-R., M.L.-C. and R.M.; supervision, M.L.-C.; project administration, M.L.-C. and R.M.; funding acquisition, R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This material is in part based upon work supported by the National Science Foundation under Grant No. 2318597. A.V.N.-R. was supported in part by the Formative Academic Experiences Program (PEAF) of University of Puerto Rico at Rio Piedras (UPRRP). This material is in part based upon work supported by the National Science Foundation under Grant No. 2030188.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data and code from experiments can be found here: https://github.com/labllluberes/beeClassificationMethods (accessed on 10 April 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Confusion matrices for event classification in the manually annotated dataset of 1194 events using `THRS_BIDIRECTIONAL`, `DISP_BIDIRECTIONAL` and `ANGLE_BIDIRECTIONAL`.

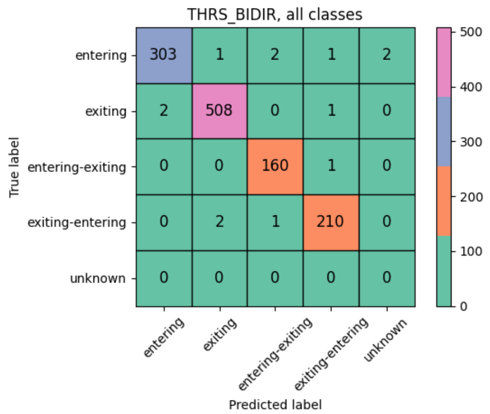


Figure A1. Event classification confusion matrix for `THRS_BIDIRECTIONAL`.

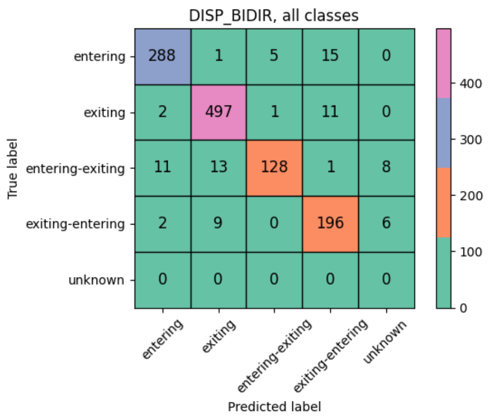


Figure A2. Event classification confusion matrix for `DISP_BIDIRECTIONAL`.

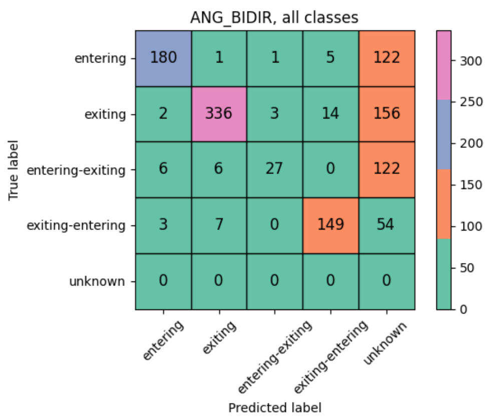


Figure A3. Event classification confusion matrix for `ANGLE_BIDIRECTIONAL`.

References

1. Garibaldi, L.A.; Steffan-Dewenter, I.; Winfree, R.; Aizen, M.A.; Bommarco, R.; Cunningham, S.A.; Kremen, C.; Carvalheiro, L.G.; Harder, L.D.; Afik, O.; et al. Wild pollinators enhance fruit set of crops regardless of honey bee abundance. *Science* **2013**, *339*, 1608–1611. [CrossRef]
2. Kulyukin, V.A.; Coster, D.; Kulyukin, A.V.; Meikle, W.; Weiss, M. Discrete time series forecasting of hive weight, In-Hive temperature, and hive entrance traffic in Non-Invasive monitoring of managed honey bee colonies: Part I. *Sensors* **2024**, *24*, 6433. [CrossRef]
3. Padubidri, C.; Kamilaris, A.; Charalambous, A.; Lanitis, A.; Constantinides, M. The be-hive project—Counting bee traffic based on deep learning and pose estimation. In *Proceedings of the SAI Intelligent Systems Conference*; Springer: Amsterdam, The Netherlands, 2023; pp. 531–545.
4. Rigakis, I.; Potamitis, I.; Tatlas, N.A.; Psirofonia, G.; Tzagaraki, E.; Alissandrakis, E. A low-cost, low-power, multisensory device and multivariable time series prediction for beehive health monitoring. *Sensors* **2023**, *23*, 1407. [CrossRef]
5. Magnier, B.; Ekszterowicz, G.; Laurent, J.; Rival, M.; Pfister, F. Bee hive traffic monitoring by tracking bee flight paths. In Proceedings of the VISIGRAPP 2018-13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Funchal, Portugal, 27–29 January 2018; pp. 563–571.
6. Colin, T.; Warren, R.J.; Quarrell, S.R.; Allen, G.R.; Barron, A.B. Evaluating the foraging performance of individual honey bees in different environments with automated field RFID systems. *Ecosphere* **2022**, *13*, e4088. [CrossRef]
7. Alburaki, M.; Madella, S.; Corona, M. RFID technology serving honey bee research: A comprehensive description of a 32-antenna system to study honey bee and queen behavior. *Appl. Syst. Innov.* **2021**, *4*, 88. [CrossRef]
8. Perry, C.J.; Søvik, E.; Myerscough, M.R.; Barron, A.B. Rapid behavioral maturation accelerates failure of stressed honey bee colonies. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, 3427–3432. [CrossRef]
9. Rodriguez, I.F.; Chan, J.; Alvarez Rios, M.; Branson, K.; Agosto-Rivera, J.L.; Giray, T.; Mégret, R. Automated video monitoring of unmarked and marked honey bees at the hive entrance. *Front. Comput. Sci.* **2022**, *3*, 769338. [CrossRef]
10. Sledevič, T.; Serackis, A.; Matuzevičius, D.; Plonis, D.; Vdoviak, G. Visual recognition of honeybee behavior patterns at the hive entrance. *PLoS ONE* **2025**, *20*, e0318401. [CrossRef]
11. Nunes-Silva, P.; Hrncir, M.; Guimarães, J.; Arruda, H.; Costa, L.; Pessin, G.; Siqueira, J.; De Souza, P.; Imperatriz-Fonseca, V. Applications of RFID technology on the study of bees. *Insectes Sociaux* **2019**, *66*, 15–24. [CrossRef]
12. Stanley, D.A.; Russell, A.L.; Morrison, S.J.; Rogers, C.; Raine, N.E. Investigating the impacts of field-realistic exposure to a neonicotinoid pesticide on bumblebee foraging, homing ability and colony growth. *J. Appl. Ecol.* **2016**, *53*, 1440–1449. [CrossRef]
13. Tenczar, P.; Lutz, C.C.; Rao, V.D.; Goldenfeld, N.; Robinson, G.E. Automated monitoring reveals extreme interindividual variation and plasticity in honeybee foraging activity levels. *Anim. Behav.* **2014**, *95*, 41–48. [CrossRef]
14. Van Geystelen, A.; Benaets, K.; de Graaf, D.C.; Larmuseau, M.; Wenseleers, T. Track-a-Forager: A program for the automated analysis of RFID tracking data to reconstruct foraging behaviour. *Insectes Sociaux* **2016**, *63*, 175–183. [CrossRef]
15. Benaets, K.; Van Geystelen, A.; Cardoen, D.; De Smet, L.; de Graaf, D.C.; Schoofs, L.; Larmuseau, M.H.; Brettell, L.E.; Martin, S.J.; Wenseleers, T. Covert deformed wing virus infections have long-term deleterious effects on honeybee foraging and survival. *Proc. R. Soc. B Biol. Sci.* **2017**, *284*, 20162149. [CrossRef] [PubMed]
16. Dicke, K.; Cassano, J.; Naug, D. Activity based performance tradeoffs differ between elite and non-elite foragers in honeybees. *Behav. Ecol. Sociobiol.* **2025**, *79*, 124. [CrossRef]
17. Requier, F.; Henry, M.; Decourtye, A.; Brun, F.; Aupinel, P.; Rebaudo, F.; Bretagnolle, V. Measuring ontogenetic shifts in central-place foragers: A case study with honeybees. *J. Anim. Ecol.* **2020**, *89*, 1860–1871. [CrossRef] [PubMed]
18. Poidatz, J.; Monceau, K.; Bonnard, O.; Thiéry, D. Activity rhythm and action range of workers of the invasive hornet predator of honeybees Vespa velutina, measured by radio frequency identification tags. *Ecol. Evol.* **2018**, *8*, 7588–7598. [CrossRef]
19. Odemer, R. Approaches, challenges and recent advances in automated bee counting devices: A review. *Ann. Appl. Biol.* **2022**, *180*, 73–89. [CrossRef]
20. Olson, E. AprilTag: A robust and flexible visual fiducial system. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3400–3407.
21. Chan, J.; Carrión, H.; Mégret, R.; Agosto-Rivera, J.L.; Giray, T. Honeybee Re-identification in Video: New Datasets and Impact of Self-supervision. In Proceedings of the VISIGRAPP (5: VISAPP), Virtual Event, 6–8 February 2022; pp. 517–525.
22. Crall, J.D.; Gravish, N.; Mountcastle, A.M.; Combes, S.A. BEEtag: A low-cost, image-based tracking system for the study of animal behavior and locomotion. *PLoS ONE* **2015**, *10*, e0136487. [CrossRef] [PubMed]
23. Sledevič, T.; Serackis, A.; Matuzevičius, D.; Plonis, D.; Andriukaitis, D. Keypoint-Based Bee Orientation Estimation and Ramp Detection at the Hive Entrance for Bee Behavior Identification System. *Agriculture* **2024**, *14*, 1890. [CrossRef]
24. Chen, C.; Yang, E.C.; Jiang, J.A.; Lin, T.T. An imaging system for monitoring the in-and-out activity of honey bees. *Comput. Electron. Agric.* **2012**, *89*, 100–109. [CrossRef]

25. Meyers, L.; Cordero, J.R.; Bravo, C.C.; Noel, F.; Agosto-Rivera, J.; Giray, T.; Mégret, R. Towards Automatic Honey Bee Flower-Patch Assays with Paint Marking Re-Identification. *arXiv* **2023**, arXiv:2311.07407. [CrossRef]
26. Santiago-Plaza, G.; Meyers, L.; Gomez-Jaime, A.; Meléndez-Ríos, R.; Noel, F.; Agosto, J.; Giray, T.; Rodríguez-Cordero, J.; Mégret, R. Identification of honeybees with paint codes using convolutional neural networks. *Proc. Copyr.* **2024**, *2*, 772–779.
27. Gómez-Jaime, A.P.; Meyers, L.; Rodrıguez-Cordero, J.A.; Agosto-Rivera, J.L.; Giray, T.; Mégret, R. Paint Blob Detection and Decoding for Identification of Honey Bees. *Proc. Copyr.* **2025**, *2*, 242–250.
28. Bjerge, K.; Frigaard, C.E.; Mikkelsen, P.H.; Nielsen, T.H.; Misbih, M.; Kryger, P. A computer vision system to monitor the infestation level of Varroa destructor in a honeybee colony. *Comput. Electron. Agric.* **2019**, *164*, 104898. [CrossRef]
29. Brandt, S.; Penaloza-Aponte, D.; Underwood, R.M.; Bruckner, S.; DeMoras, B.; López-Uribe, M.M.; Urbina, J. BeeCam-AprilTag: Timing the flight duration of individual honey bees in remote field settings with AprilTag and Raspberry Pi. *MethodsX* **2025**, *15*, 103624. [CrossRef]
30. Penaloza-Aponte, D.; Brandt, S.; Dent, E.; Underwood, R.M.; DeMoras, B.; Bruckner, S.; López-Uribe, M.M.; Urbina, J.V. Automated entrance monitoring to investigate honey bee foraging trips using open-source wireless platform and fiducial tags. *HardwareX* **2024**, *20*, e00609. [CrossRef] [PubMed]
31. Tu, G.J.; Hansen, M.K.; Kryger, P.; Ahrendt, P. Automatic behaviour analysis system for honeybees using computer vision. *Comput. Electron. Agric.* **2016**, *122*, 10–18. [CrossRef]
32. Nhung, T.L.; Hong, P.T.T.; Lan, L.T. A method for bee activities recognition from videos captured at the beehive entrance. *J. Mil. Sci. Technol.* **2024**, *8*, 3–13. [CrossRef]
33. Mukherjee, S.; Kulyukin, V. Application of digital particle image velocimetry to insect motion: Measurement of incoming, outgoing, and lateral honeybee traffic. *Appl. Sci.* **2020**, *10*, 2042. [CrossRef]
34. Lei, C.; Lu, Y.; Xing, Z.; Zhang, J.; Li, S.; Wu, W.; Liu, S. A Honey Bee In-and-Out Counting Method Based on Multiple Object Tracking Algorithm. *Insects* **2024**, *15*, 974. [CrossRef]
35. Jocher, G.; Qiu, J. Ultralytics YOLO11. 2024. Available online: https://github.com/ultralytics/ultralytics (accessed on 20 February 2026).