Non-invasive Honeybee Colony Monitoring via Robotic Mapping of Combs in Observation Hives

Jiří Janota[a], Jan Blaha[a], Martin Stefanec[b], Tomáš Rouček[a], Jiří Ulrich[a], Laurenz Fedotoff[b], Fatemeh Rekabi-Bana[c], Farshad Arvin[c], Thomas Schmickl[b], Tomáš Krajník[a]

[a]*Artificial Intelligence Centre, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, 12000, Czechia*
[b]*Artificial Life Lab, Institute of Biology, University of Graz, Graz, 8010, Austria*
[c]*Department of Computer Science, Durham University, Durham, DH1 3LE, UK*

Abstract

Honeybees play a crucial role in our ecosystem, as their pollination efforts directly affect the global food web and biodiversity of flowering plants. Consequently, in light of the worldwide decline of pollinators, understanding the dynamics of honeybee colonies and monitoring their health attracted high research interest. One possible approach to health assessments is to estimate the number of brood and food storage cells in the honeybee comb. However, the cells are difficult to observe directly due to occlusions caused by the worker bees. This requires either manual removal of the comb and brushing off the bees, or continuous monitoring of the entire comb. In this work, we present a method for automated monitoring of brood cells over long time periods by a robot operating in observation hive. Our approach uses focused honeybee queen observations to detect egg-laying events and opportunistic and irregular image scans to detect and classify individual cells in the comb. By combining these modules with a temporal model of brood development, we create a spatially and temporally consistent semantic map of the comb. The resulting map allows us to continuously estimate the total numbers of different brood cell types in the comb, as well as the brood cannibalism rate. Although the robotic platform used in this study is not applicable to standard hives, our work contributes valuable data for modeling honeybee colonies and understanding the structural dynamics of the comb.

Keywords: Apis mellifera, biohybrid systems, computer vision, mapping, deep learning, long-term autonomy

1. Introduction

Western honeybees (*Apis mellifera*) play an important role in our ecosystem, as their pollination efforts directly influence biodiversity and food security Paudel et al. (2015); Shaheen et al. (2017); Allsopp et al. (2008). In light of the worldwide decline of pollinators Potts et al. (2010); Kevan and Viana (2003), researchers have in recent years been investigating ways to support honeybee colonies with technology. Specifically, advances in robotics and artificial intelligence opened up new opportunities in honeybee research for animal-robot interaction in natural ecosystems Barmak et al. (2023); Romano et al. (2024).

To continuously assess the health of the colony, one would like to monitor the entire honeybee comb, which encodes information about the strength of the colony and also about the outside

environment. The comb reveals the critical health indicators of the colony, such as the state of food stores (nectar, honey, and pollen), the growth of the colony (through the development from eggs to adult bees), and potential issues such as diseases or cannibalism of the brood Groeneveld et al. (2024). Historically, comb inspections were performed manually, experienced observers estimating the number of honeybees in the comb, as well as the quantity of cells for food storage and brood. However, such frequent manual inspections are time-consuming and require trained observers, which makes them unscalable and subject to potential bias. Moreover, manual inspections can also disrupt colony interactions and carry the risk of harming or killing the queen or workers.

To alleviate these issues, some researchers have attempted to automate parts of the inspection process to reduce workload and improve accuracy (see Section 2). However, most existing methods are still invasive, necessitating the removal of honeybee combs from the hive and the removal of bees (usually with a brush), which disrupts the colony. Unfortunately, this is still hardly scalable for continuous monitoring of a large number of colonies. Moreover, our goal is to observe the colony in its natural environment while minimising disturbances; traditional methods are, therefore, unsuitable.

In addition to its relevance in biological and agricultural research, our work is also instrumental in our two Horizon projects "RoboRoyale" and "SensorBees". The first aims to support the efficiency and growth of honeybee colonies through interactions with a honeybee queen, who is practically responsible for the colony's growth. The second seeks to use honeybee colonies as a sensor networks for environmental surveillance through observation of the comb contents.

In the RoboRoyale project, we designed a robotic system capable to observe and interact with a living honeybee colony with a specific focus on the honeybee queen, as outlined in Ulrich et al. (2024). When the queen is inactive, such as during sleep, the robotic system can opportunistically scan the comb, producing a grid of partially overlapping images captured at different locations to provide comprehensive coverage of the entire comb to capture the brood state. Detailed knowledge of the brood is essential to assess the impact of robotic interventions and interactions on the whole colony.

In this paper, we present a software pipeline that uses comb scans, irregular in time, to track honeybee brood development at the individual cell level. Our approach leverages robotic mapping techniques to create a semantic map of the comb, where each cell is modelled independently. We integrate multiple detectors for egg-laying activity, open brood, and capped brood with a temporal model that captures the stages of honeybee development. Although automated data collection often results in images where cell contents are not clearly visible, we demonstrate that neural networks can learn patterns beyond human annotators' capabilities.

Tracking individual cells over time enables estimating the number of egg, larva, and capped brood cells, as well as the state of unobserved cells and the age and sex of larvae and capped brood. Our approach provides unprecedented real-time comb monitoring in observation hive without disrupting the colony and without the need for manual intervention. Since our model captures standard colony development, it can also be used to detect anomalies, investigate broodnest formation, and quantify brood cannibalism, providing potentially valuable insights into colony health. While this system is still operating only on observation hives and therefore is limited in scalability, it provides a useful proof-of-concept solution for studying honeybees in partially controlled conditions and methods of long-term monitoring.

1.1. Contributions of the Work

The main technical contribution of this work is a mapping pipeline able to deal with a cluttered and repetitive environment, which exhibits only sparse features with limited correspondences. An important part of the pipeline is a temporal filter integrating a model of brood development and three different visual sensors. The filter enables us to monitor the state of the colony even with infrequent and irregular comb observations and a high level of occlusions caused by honeybees. A minor contribution lies in showing that, using an intelligent annotation scheme, we were able to leverage temporal context in order to learn a model to classify images not annotatable by humans. A major biological contribution is the creation of the first complete map of individual comb brood cell states' evolution over an extended time period. We believe this to be the first effort to model the honeybee cell development using a temporal Bayes filter and incorporate the queen behaviour into the cell state estimation. The presented approach shows that the underlying theoretical foundation of robotic mapping methods, e.g., Bayesian filtering, is appropriate and applicable to comb state estimation.

2. Related Works

Researchers have studied honeybee colonies and methods to estimate their strength for decades. Traditionally, colony health was assessed manually, relying on direct counts of eggs, brood, or adult worker bees. Jeffree (1951) presented an estimation method that compared the comb images in each hive with the reference images, allowing for efficient population assessments in hives. In Jeffree (1958), a method for estimating brood areas using a grid placed over the brood frames was introduced. These works laid the foundations for the widely used Liebefeld method, which relies on visual estimation of the number of adult workers covering each side of a comb, as well as the surface area occupied by open brood, capped brood, honey and pollen stores Dainat et al. (2020). This method standardized bee population and cell type quantification estimates across different colonies.

With advancements in technology, researchers have increasingly turned to computer-assisted methods to ease colony assessments and improve their accuracy and efficiency. The majority of existing non-invasive monitoring solutions rely on indirect measurements such as weight, temperature, humidity, sound, or bee traffic Meikle and Holst (2014); Hadjur et al. (2022). Although these measurements can reflect overall state of the colony, they provide no direct information about the condition of the comb itself. To address this, visual-based approaches have been developed to analyze comb images directly. Most studies have focused on estimating the brood area or counting the number of cells within the honeybee comb. The early methods, such as Emsen (2006) and Yoshiyama et al. (2011), still required a human operator to segment images manually and did not offer any automatic analysis.

Knauer et al. (2005) was among the first to propose an automated approach for detecting open cells in comb images partially occluded by bees, using an adaptive background model, edge detection and contour classification. Liew et al. (2010) introduced a method to detect and count individual cells in comb images without bees, employing the Circle Hough Transform (CHT), a widely used technique for circular feature detection Illingworth and Kittler (1987). Höferlin et al. (2013) further advanced this technique by integrating machine learning algorithms for cell classification based on their content and offering their solution through the commercial software "HiveAnalyzer".

Other researchers explored alternative methods for cell detection in images without any bees. For instance, Sparavigna (2016) used thresholding and superpixel partitioning to evaluate cell

size distributions, while Rodrigues et al. (2016) applied a convolution-based approach to cell detection with subsequent classification of capped brood cells. Despite these efforts, CHT remained the most prevalent approach for cell detection. Tools such as "CombCount" Colin et al. (2018) utilised CHT to automatically detect open cells and accelerate manual annotation of areas with capped cells (brood and honey areas). Similarly, studies such as Paolillo et al. (2022) and Rathore et al. (2023) employed CHT to count uncapped cells on combs as well. In Alves et al. (2020), the authors used CHT for cell detection and combined it with a U-Net neural network to better identify the area of the comb and filter out false positive detections. They also compared various neural networks for cell classification based on cell contents, presenting their results as the "DeepBee" software. Rodriguez-Lozano et al. (2024) addressed the problem of detecting capped honey cells, which usually may not have clear circular borders. Then high-resolution comb images were split into tiles analysed by a neural network to segment areas with capped honey. Nevertheless, all of the mentioned visual-based semi-automatic methods are invasive and require the removal of the bees from the comb, thus causing disruption to the colony.

In some works, the authors use a fixed-camera setup to observe the honeybee comb with the colony over longer periods. In Bozek et al. (2021), the authors trained a U-Net neural network to detect capped brood cells in a background-filtered image and combined it with bee tracking to estimate the demographic population of the colony. Another work, Defèr (2022), took a different approach and used a convolutional neural network to localize centres of open cells not occluded by bees and attempted to estimate larvae age from the images using deep learning.

Recently, a few alternative methods have been proposed to observe the comb in novel ways. Borlinghaus et al. (2024); Stefanec et al. (2025) employed a modified flatbed scanner, enabling detailed tracking of individual cell brood development from egg to adult bee. Notably, this setup allows observation inside sealed cells and has made it possible to directly capture the reproductive cycle of the Varroa mite within them. In contrast, Milovanović et al. (2025) introduced "Beeholder", a system that embeds sensors into standard hive frames to analyze comb contents via absorption spectroscopy. Their results demonstrate the ability to detect various patterns of honey filling. While it does not offer cell-level resolution yet, its design allows continuous comb monitoring without disturbing the colony.

The most comprehensive commercial attempt to automate beekeeping seems to be the "Beewise" system Beewise Technologies Ltd. (2025). According to the propagation materials, it employs a custom robotic system, which is capable of automatically carrying out tasks such as analyzing the comb contents using visual-based methods, applying treatments, or harvesting honey. This enables routine, full-frame inspections of all combs in the hive without human intervention, but at the cost of colony disturbance due to repeated frame manipulations. However, it is important to note that as it is a commercial product, very little information is available as to its capabilities and inner design.

We build on our previous work Janota et al. (2024), in which we already work with data from a robot with a moving camera setup by Ulrich et al. (2024). There, we demonstrated the application of standard object detection neural networks for cell detection in images occluded by bees. We have also shown the limitations of standard image registration methods when used in the dynamic, repetitive, and cluttered environment of a honeybee hive. Additionally, we demonstrated that with sufficiently precise positional information of the robot, we can gather observations of individual cells over time. Unlike in the works of our predecessors, our observations are local. This necessitates to employ data sampling strategies capable to deal with occlusions caused by worker bees Blaha et al. (2024). In this work, we propose a robust method for creating a spatial semantic map of the comb, enabling detailed long-term tracking of the state of individual brood

cells in the comb.

We provide a concise comparison of our work with several previously mentioned open source or commercial solutions in Tab. 1. We compare the approaches based on the level of autonomy (i.e., whether human involvement is required for operation), invasiveness (i.e., whether it needs to physically manipulate with the comb frames), cell-level resolution (i.e., whether it provides aggregated data or detailed information about individual cells), monitoring method, whether it targets standard beekeeping practices and is applicable to field-size hives and colonies, and product availability.

Feature	Beewise*	Beeholder	DeepBee	Ours
Fully Autonomous	✓	✓	✗	✓
Non-invasive	✗	✓	✗	✓
Cell-level Resolution	✓	✗	✓	✓
Monitoring Method	Visual	Absorp. spectroscopy	Visual	Visual
Targets Standard Beekeeping	✓	✓	✓	✗
Product Availability	✓	✓	✓	✗

Table 1: Comparison of our mapping system, which uses a vertical gantry robotic system for data acquisition, with several commercially available solutions for comb monitoring. *Note that the information in this table may be incomplete or misleading, as there is very little information available for the commercial system "Beewise".

3. System Setup and Data Acquisition

Our experimental setup consists of an observation hive containing two vertically stacked combs of standardised size 420 mm × 220 mm, covered by glass panels.

The observation hive is constructed from aluminium extrusion parts to ensure high-precision, comparable and reproducible observations. The hive features two glass panes on each side, which permit 100% visibility of the comb surfaces in principle. The internal design of the hive provides a volume of around 9 litres, with a narrow depth of around 4 cm between the glass panes for each comb viewing area. It also includes air slits for ventilation and a port for a feeder. The total internal volume, coupled with the use of only two combs, means that the hive was significantly smaller than typical field production hives, thereby accommodating a smaller colony size than is generally found in such hives. Observational hive systems, including the type utilised in this study, generally have a spatial volume that is insufficient for honeybee colonies to achieve their full biological size. In biological terms, the limited comb area in such hives is rarely sufficient for a colony to reach its maximum population size. We estimate that these confined conditions typically allow colonies to reach a maximum size of around 15% of what they might attain in larger, production-oriented hives. These significant constraints on overall colony size inherently influence natural developmental trajectories. While this setup allows for detailed observation, it represents a specific experimental context. The hive is housed at the University of Graz in Austria, indoors to facilitate hardware setup and protect the glass-covered hive from direct sunlight. A plastic tube connects the observation hive, which is indoors, to the outdoor environment allowing for free movement of the bees. To minimize disturbance to the colony, observations are conducted under near-infrared LED lighting with wavelengths from 750 nm to 850 nm, which is invisible to honeybees. The hive is located on the third floor of a university building and kept in an indoor room under constant darkness (DD conditions) for the bees. The hive is kept in a climate-controlled indoor room with typical ambient temperatures

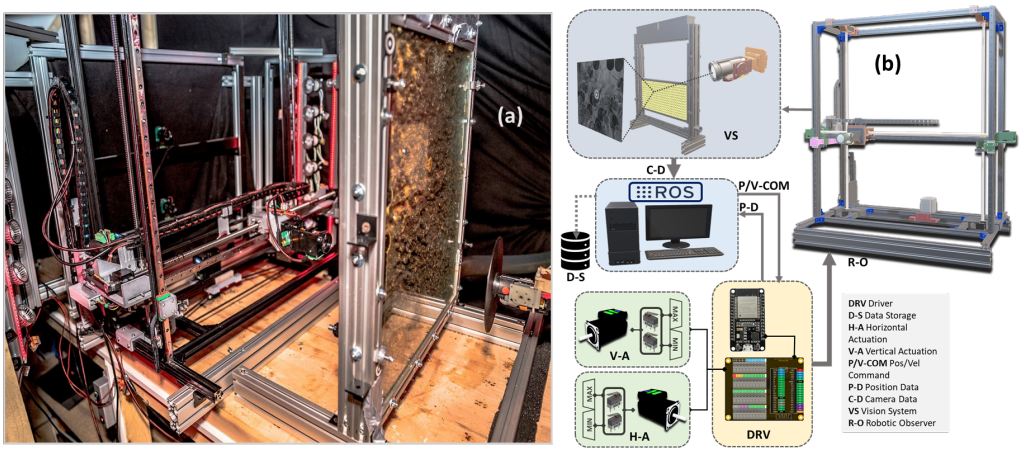


Figure 1: The autonomous observation system collecting behavioral information of the queen honeybee. Panel (a) shows a photo of the system in its real workspace with the observation hive, and panel (b) shows the mechatronic system and its main components for data collection.

ranging from 26 °C to 30 °C. This indoor setting results in relatively stable temperatures with less daily fluctuation compared to typical field conditions, as there was no direct sunlight exposure on the hive. While relative humidity is not actively controlled, it is comparatively low due to the air conditioning system in use. Despite the indoor housing of the hive itself, the bees have continuous access to the outside via a flight tube, allowing them to forage freely. The external foraging environment is distinctly urban, situated within the city. It offers diverse floral resources from several key areas in the vicinity, including a nearby botanical garden, the central city park, and numerous tree-lined avenues and urban green spaces providing a variety of flowering plants. When natural forage was insufficient, a 72.7% (w/w) sucrose solution was provided *ad libitum* to ensure adequate nutrition.

To achieve long-term autonomous colony monitoring, we use a vertical robotic gantry system, AROBA (see Fig. 1), introduced in Ulrich et al. (2024). The robot has two independent ball screw drive systems for movement in both horizontal and vertical directions, supported by two linear guides for stability. A stepper servo motor, with a resolution of about 4 µm and maximum speed of 10 mm s^{-1}, drives a single ball screw for horizontal motion. For vertical motion, two synchronized ball screw drives, powered by a stepper servo motor, work together to improve movement stability and precision. For a full coverage of the observation hive, we use two such mechanisms working in parallel, each at one side of the comb, see Fig. 1. Further details about the robotic system can be found in Ulrich et al. (2024).

3.1. Vision System

The robotic system is equipped with a high-resolution Active Silicon Harrier 10x AF-Zoom camera mounted on its end-effector that can be positioned parallel to the comb to observe any location in the hive. The camera uses the Sony Starvis sensor for high performance in low-light conditions. The camera captures images at a resolution of 1920 px × 1080 px at a rate of 30 Hz and features controllable zoom and focus. We collect images at two zoom levels—queen tracking and searching are performed at a resolution of approximately 67 µm px^{-1}, while

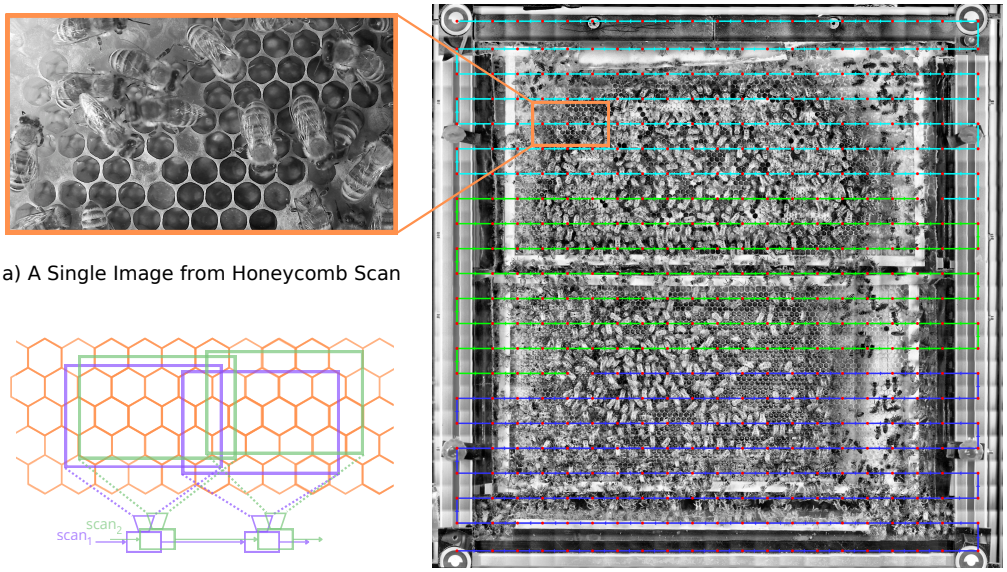


Figure 2: A visualization of the scanning process, where the robot sequentially moves through the comb, capturing images at prespecified locations. Panel a) shows a single image observation of the comb (with its location highlighted in orange color). Panel b) shows a diagram of the scanning data with image alignment needed in both in space (green–green, purple–purple) and time (green–purple). Panel c) shows an image of the entire comb stitched from observations at individual locations (red dots), created by merging three partial scans taken at different times (highlighted in different colors).

comb observations are conducted at a resolution of approximately $37\,\mu\mathrm{m\,px}^{-1}$. For tracking the honeybee queen, we use a marker-based vision system, presented in Ulrich et al. (2023).

3.2. Scanning the Honeycomb

The robotic mechanisms on both sides of the hive operate in parallel, with one robot actively tracking the queen while the other mirrors its movement. This ensures continuous observation even when the queen moves to the opposite side of the hive. When the queen is resting, the secondary robot, which normally mirrors the tracking one, switches to scanning the comb. The scan begins at the lower-left corner and progresses sequentially, with the robot stopping at pre-specified locations in the comb and capturing images. If the queen starts moving, the scan is preempted and later resumes from the last recorded scan position.

This process generates partial scans of the comb, consisting of a series of overlapping images. By sequentially combining these partial scans—starting from the initial scan at the lower-left corner to the scan ending at the upper-left corner of the comb—we can construct a complete scan of the comb (see Fig. 2). The complete scan forms a grid of a total of 462 overlapping image tiles, each captured at a different location in the hive.

3.3. Odometry Calculation

To create a map of the comb, it's crucial to know the location of the robot's end-effector within the hive. The robotic system generates position information from the motors, though this data

can be inconsistent over time. In Ulrich et al. (2024), the evaluation of the system showed less than 8 mm drift after traversing a 1 km distance.

To mitigate this, the AROBA system applies a transformation of the motor positions into the hive's coordinate system. This transformation relies on markers placed at each corner of the hive, as shown in Fig. 2. During the calibration process, the robotic manipulator moves to the centre of each marker, recording the motor positions at each location. A homography matrix is computed to project the motor positions onto the hive's coordinate frame. The robotic manipulator is then navigated in the coordinate frame of the hive. Before every start of the scanning, the robot moves to the first marker and compensates the accumulated drift so that the position of the first marker is always in the origin of the hive's coordinate frame.

3.4. Produced Data

We use a dataset collected from 15 August 2024 to 9 September 2024, consisting of honeybee queen tracks and scans of the comb. In total, we tracked the honeybee queen for 58.1% of time, with the total traveled distance being 849.8 m. On the side of the hive labelled 0, the robotic system collected 73 full comb scans and 103 full comb scans on side 1.

4. Methods

In this paper, we propose a method for mapping a honeybee comb using data produced by the robotic system described in Section 3. The goal is to create a spatially and temporally consistent semantic map of the comb, with individual cells being the atomic map element.

4.1. Problem Statement

The mapping process operates with two kinds of inputs—comb scans and queen tracking data. We define the operational space of the robot as $\mathcal{A} \subset \mathbb{R}^2$, representing the full range of motion available to the robotic system. This operational space includes and exceeds the comb area–the area physically occupied by two vertically stacked combs, each measuring 420 mm × 220 mm–thus enabling the robot to observe any location within the hive (see Fig. 2c). The sequence of all collected full comb scans is denoted as $(S_s)_{s=1}^{n}$, where n is the total number of scans. Each full scan S_s consists of a sequence of 462 comb observations $((\mathbf{p}_{s,j}, t_{s,j}, \mathbf{I}_{s,j}))_{j=1}^{462}$, where $\mathbf{p}_{s,j} \in \mathcal{A}$ is the camera position in the hive's coordinate frame, $t_{s,j} \in \mathbb{R}^+$ represents the timestamp the observation was captured and $\mathbf{I}_{s,j}$ is the image tile. In our particular setup, the total number of tiles is always 462, see Section 3.2 for details.

The queen tracking data consists of an image stream from the system when it is tracking the queen. Individual time-continuous data are denoted by a sequence $(Q_q)_{k=1}^{m}$. Each queen track Q_q consists of a sequence of observations $((\mathbf{p}_{q,j}, t_{q,j}, \mathbf{I}_{q,j}))_{j=1}^{L_q}$, where $L_q \in \mathbb{N}$ is the length of the track, $\mathbf{p}_{q,j} \in \mathcal{A}$ is the position of the camera in the hive coordinate system, $t_{q,j} \in \mathbb{R}^+$ is the timestamp of the honeybee queen observation, and $\mathbf{I}_{q,j}$ is the corresponding image of the honeybee queen.

The goal is to generate a sequence of semantic maps $(M_s)_{s=20}^{n}$, where each map M_s integrates information from the scans and honeybee queen tracks collected up to that point. The semantic map is represented as a set of individual cells $M_s = \{c_{s,1}, \ldots, c_{s,N_s}\}$, where each cell $c_{s,j}$ is characterized by its metric position $\mathbf{p}_{s,j} \in \mathcal{A}$, its estimated radius $r_{s,j} \in \mathbb{R}^+$, number of observations of the cell $o_{s,j} \in \mathbb{N}$, its state $\mathbf{s}_{s,j} \in [0, 1]^{43}$ representing belief over possible cell contents, and the last timestamp $t_{s,j} \in \mathbb{R}^+$ when the cell was observed. We define 4 content classes $k \in \{egg, larva, capped\ brood, other\}$, with a total of 43 internal states, see Section 4.4.1.

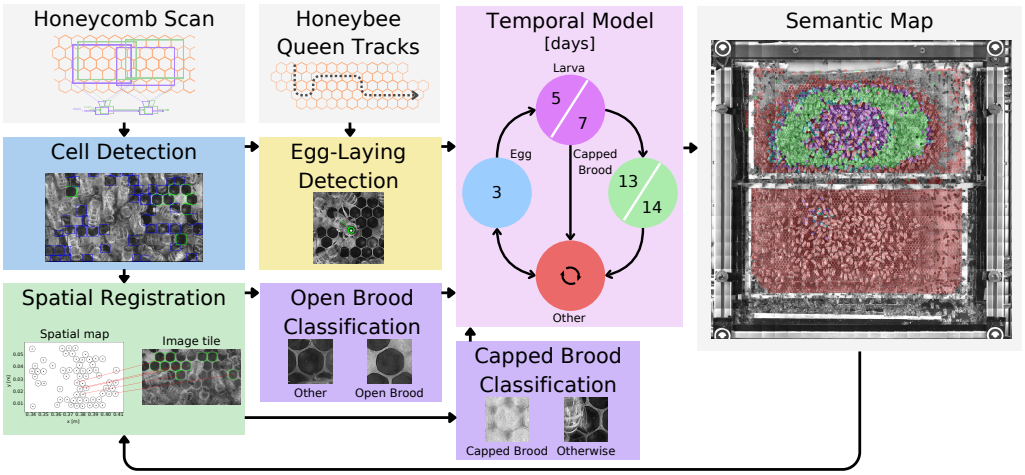


Figure 3: Overview of the mapping pipeline. First, data from the honeybee comb scans are used to build the map of the spatial layout by detecting open cells and registering the observations. Semantic information is extracted from the scan images and from honeybee queen tracking. Cell detections are classified into open-brood and non-brood classes, for cells where capping is expected also capping detector is in place. Simultaneously, honeybee queen tracks are analyzed to detect egg-laying events. All the indirect observations are passed to a Bayesian filter, which encodes the temporal development of brood stages, producing the final spatially and temporally consistent semantic map including age estimates.

It is important to note that with new scans, cells may not only be added but also removed from the map, which is why $|M_s|$ can be less or more than $|M_{s+1}|$. To track the evolution of individual cells across scans, we need to ensure that $\mathbf{s}_{s,j}$ integrates only observations corresponding to the same cell. For this purpose, we could either characterize each map cell with an additional unique identifier or define a mapping between two maps M_s and M_{s+1}. We define the mapping as a matrix $A_{s,s+1} \in \{0,1\}^{N_s \times N_{s+1}}$, where $A_{s,s+1}(i,j) = \mathbb{1}[c_{s,i}$ corresponds to $c_{s+1,j}]$.

4.2. Approach

An overview of the proposed system is shown in Fig. 3. As mentioned, the mapping process takes as input tracks of the honeybee queen, sequential scans of the honeybee comb, and, once available, a previously constructed map. The whole mapping process can be thought of as two alternating steps: (1) mapping of the spatial layout (Section 4.3) and (2) extending the map with semantic information (Section 4.4).

At the beginning of the spatial mapping process, when the previous map is not available, we use the first n comb scans $(S_s)_{s=1}^{n}$ to create an initial spatial map M_n (see Section 4.3.3), which is then further updated as new data becomes available (Section 4.3.4). The map update process begins with the detection of open cells in the collected image tiles $(\mathbf{I}_{s,j})_{j=1}^{462}$ in the full honeybee comb scan S_s (Section 4.3.1). The detected cells are then processed by a spatial registration module (Section 4.3.2), which establishes correspondences between the cells already in the map and the detected ones and updates the spatial map accordingly.

After having integrated cell detections into a spatially coherent structure (both initial map creation and consequent map updates), we also integrate the semantic information based on the cell content. Unfortunately, the lighting conditions and amount of debris in the hive prevent us from directly and easily observing the contents, so we work with three different kinds of observations,

which sometimes partially overlap with the set of cell states, however, in general, we consider these hidden. First, we classify the images of detected cells using the open brood classification module to later discern cells with eggs and larvae (Section 4.4.2). Cells that are already present in the map but remain undetected in the latest scan are processed by the capped brood classification module, which looks at their assumed positions and confirms the larva capping (Section 4.4.3). To further improve the cell state estimation, the system utilizes queen tracking data to detect egg-laying events, indicating which cells may contain a newly laid egg (Section 4.4.4). Finally, the outputs from the classification modules and egg-laying detection associated with cell $c_{s,j}$ are integrated into the state estimate $\mathbf{s}_{s,j}$ using a temporal Bayes filter, which continuously estimates and updates the state of each cell in the map (Section 4.4.1). For simplification, in our approach, we adopt the assumption of independence between the states of cells, as is common in similar mapping methods.

4.3. Spatial Mapping

In this section, we explain how the underlying spatial map is created and updated. As mentioned, the spatial map is represented as a set of individual cells $M_s = \{c_{s,1}, \ldots, c_{s,N_s}\}$, each characterized by their estimated metric position $\mathbf{p}_{s,j} \in \mathbb{R}^2$, radius $r_{s,j} \in \mathbb{R}^+$ and number of detections of the cell $o_{s,j} \in \mathbb{R}^+$. The mapping starts with generating an initial map $M_{n_{\text{init}}}$, using the first n_{init} scans $(S_s)_{s=1}^{n_{\text{init}}}$ of the honeybee comb, where we chose $n_{\text{init}} = 20$ (Section 4.3.3) and the subsequent scans are used to continuously update the map (Section 4.3.4).

4.3.1. Honeybee Cell Detection

The first step in constructing the map is detecting individual open cells in honeybee comb scan images, as they serve as the foundation for subsequent spatial registration. The detection of open cells in honeybee comb images has been widely studied, as discussed in Section 2, with the Circle Hough Transform (CHT) being the most commonly used approach. However, in our previous work Janota et al. (2024), we demonstrated that CHT struggles when used on images from live colonies, leading to a high number of false positive detections. To overcome these limitations, we proposed using standard object detection neural networks, specifically Faster R-CNN Ren et al. (2016) and YOLOv5 Jocher et al. (2022).

The cell detection process is performed on individual image tiles of size 1920 px × 1080 px extracted from honeybee comb scans. With the aim to enhance the neural networks' ability to detect open cells that are not occluded by bees, we introduce two distinct classes: *fully visible* open cells and *partially occluded* open cells. For a comprehensive evaluation, we reassess these methods on our new annotated dataset, as detailed in Section 5.1. The training process of the object detection methods, along with the specific CHT parameters, is described in our previous work Janota et al. (2024).

The input to the honeybee cell detection is a single image tile, $\mathbf{I}_{s,j}$, with its corresponding metric position, $\mathbf{p}_{s,j} \in \mathbb{R}^2$. The detector outputs bounding boxes with class labels. For spatial mapping, we use only *fully visible* detections, estimating each cell's metric position using the center of its bounding box and computing its radius as the average of the bounding box's vertical and horizontal sizes.

4.3.2. Cell Detection-based Registration

To be able to create the initial map and perform map updates, correspondences between the open cell detections in the comb images and also between the open cell detections and the existing map need to be established. There are two main approaches when registering two images:

direct approach and feature-based approach. In our previous work Janota et al. (2024), we evaluated both approaches and showed their limitations. The comb images exhibit variations in illumination and exposure, and the overlapping parts of the images usually differ significantly in content due to the movement of the honeybees, making the image registration challenging for the direct approach based on cross-correlation. Conversely, there is a lack of unique features for standard feature-based approaches due to the repetitive nature of the honeybee comb.

To address these challenges, we employ a feature-based registration using the detected open cells as unique features. If the robot's odometry is sufficiently precise—i.e., with a position error smaller than half the cell size—the registration based on detected cells, which minimizes the translation between the positions of corresponding cells, is sufficient. As demonstrated in Section 5.2, this condition holds in our case. Additionally, this registration method can also be used to register new comb scans to an existing map.

In both position-based registration of image pairs and image-to-map registration, the problem simplifies to aligning two point sets $\mathcal{D}_1 = \{\mathbf{d}_1, \ldots, \mathbf{d}_m\} \subset \mathcal{A}$ and $\mathcal{D}_2 = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\} \subset \mathcal{A}$, where each point represents the position of a detected cell or a cell in the existing map. Since the robot moves in a plane parallel to the honeybee comb, the transformation between the point sets is restricted to translation along the horizontal and vertical axes. To estimate this transformation, we employ a sampling-based algorithm that searches for a minimal translation between the point sets, to identify corresponding cells $C^* = \{(\mathbf{d}, \mathbf{d}') \mid \mathbf{d} \in \mathcal{D}_2, \mathbf{d}' \in \mathcal{D}_1\}$ and determine the translation vector $V^* \in \mathbb{R}^2$. We define the distance threshold G, set to the typical cell size about 4.4 mm (varies with camera placement), which serves as an initial search radius. $\frac{G}{2}$ then provides a stricter criterion for final correspondences. The pseudocode for the cell detection-based registration is presented as Algorithm 1.

4.3.3. Creating Initial Map

When the odometry of the robot is imprecise or cell detections are sparse, the incremental approach to map construction may fail. To address this, we designed a more robust technique for generating the initial map. This process consists of two phases: (1) constructing per-tile maps by aligning images captured from the same locations in the hive and (2) merging these per-tile maps into a complete initial map. As already mentioned, we use the first n_{init} comb scans $(S_s)_{s=1}^{n_{\text{init}}}$ to create the initial map $M_{n_{\text{init}}}$, in order to increase robustness of the mapping process. We selected $n_{\text{init}} = 20$ as empirically reasonable, but we observed the method not being too sensitive to the exact choice of n_{init} on our data. The sensitivity of the method to n_{init} is related to the precision of the robot's odometry, with less precise odometry needing a better initial map (see Sec. 5.2 for information on the precision of our robot).

Creating Per-tile Maps. The process begins by grouping images $(\mathbf{I}_{s,k})_{s=1}^{20}$ taken from the same location $k \in \{1, \ldots, 462\}$ in the hive and applying cell detection-based registration to establish correspondences $C_{i,j} = \{(\mathbf{d}, \mathbf{d}') \mid \mathbf{d} \in \mathcal{D}_i, \mathbf{d}' \in \mathcal{D}_j, i \in \{1, \ldots, 20\}, j \in \{1, \ldots, 20\}, i \neq j\}$ between the cell centers $\mathcal{D}_i$ and $\mathcal{D}_j$ detected in images $\mathbf{I}_{i,k}$ and $\mathbf{I}_{j,k}$ (see Fig. 4). These correspondences serve as the basis for estimating pairwise translations $\mathbf{T}_{i,j}$ between the images. Next, we anchor the first image tile $\mathbf{I}_{1,k}$ to its metric position $\mathbf{p}_{1,k} \in \mathbb{R}^2$ and refine the positions $\mathbf{p}_{s,k} \in \mathbb{R}^2$ of all other images $(\mathbf{I}_{s,k})_{s=2}^{20}$ in the sequence using a least-squares optimization constrained by the

Algorithm 1: Cell detection-based registration to find correspondences between the cells

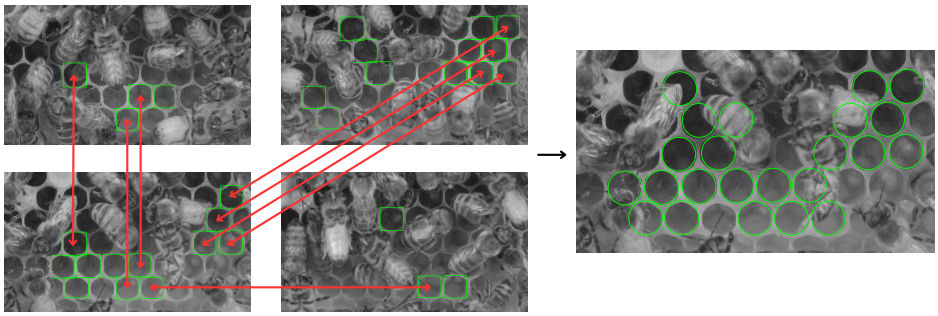
Input: *Two sets of cells positions:* $\mathcal{D}_1$, $\mathcal{D}_2$
Output: *Optimal correspondences* C^*
$S \leftarrow \{\mathbf{d} \mid \mathbf{d} \in \mathcal{D}_2, \exists \mathbf{d}' \in \mathcal{D}_1, \|\mathbf{d} - \mathbf{d}'\| < G\}$;
Shuffle(S);
$C^* \leftarrow \emptyset$;
$V^* \leftarrow \infty$;
foreach $\mathbf{d} \in S$ *until max iterations* **do**
  foreach $\mathbf{d}' \in \mathcal{D}_1$ ***where*** $\|\mathbf{d}' - \mathbf{d}\| < G$ **do**
    $\mathbf{T} \leftarrow \mathbf{d} - \mathbf{d}'$;
    $\mathcal{D}_2' \leftarrow \mathcal{D}_2 + \mathbf{T}$;
    $C \leftarrow \{(\mathbf{d}, \mathbf{d}') \mid \mathbf{d} \in \mathcal{D}_2', \exists \mathbf{d}' \in \mathcal{D}_1, \|\mathbf{d}' - \mathbf{d}\| < \frac{G}{2}\}$;
    if $|C| > 0$ **then**
      $\bar{\mathbf{T}} \leftarrow \frac{1}{|C|} \sum_{(\mathbf{d},\mathbf{d}') \in C} (\mathbf{d} - \mathbf{d}')$;
      $V \leftarrow \|\bar{\mathbf{T}}\|$;
      if $V < V^*$ **then**
        $C^* \leftarrow C$;
        $V^* \leftarrow V$;
      end
    end
  end
end

pairwise translations $\mathbf{T}_{i,j}$ between them to satisfy

$$\min_{\forall i: \mathbf{p'}_{i,k}} \sum_{\substack{i,j \in \{1,\ldots,20\} \\ i \neq j}} ((\mathbf{p'}_{i,k} - \mathbf{p'}_{j,k}) - \mathbf{T}_{i,j})^2 \quad \text{s.t.} \quad \mathbf{p'}_{1,k} = \mathbf{p}_{1,k}. \tag{1}$$

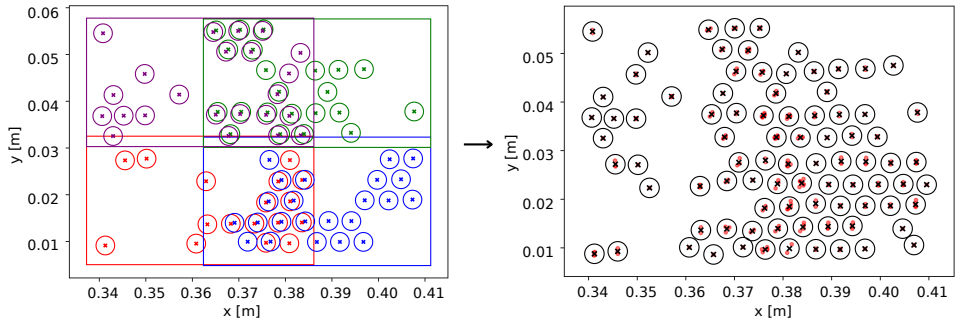
Once the images are aligned, we cluster detections from different images whose refined positions lie within $\frac{G}{2}$ of each other, with G set to the typical cell size about 4.4 mm, to associate them as detections of the same physical cell. False positive detections or false registration of the cells may result in multiple overlapping cells in the per-tile map. This is inconsistent with the physical constraints of the comb structure. To ensure spatial consistency, we apply Non-Maximum Suppression (NMS) with an Intersection over Union (IoU) threshold of 0.3. Of the overlapping cells, we retain those with a higher number of detections. The complete pseudocode for the creation of per-tile maps is presented in Algorithm 2.

Combining Per-tile Maps. After constructing the per-tile maps P_k where $k \in \{1, \ldots, 462\}$, we merge them into a comprehensive initial map of the entire comb, as illustrated in Fig. 5. We again employ cell detection-based registration (Alg. 1) to establish correspondences between neighboring per-tile maps, producing a set of unique cells in the honeybee comb. In the same fashion as with the per-tile-maps we recalculate the metric positions and radii of the cells and apply NMS, ensuring spatial consistency. The pseudocode outlining the per-tile-maps merging process is presented in Algorithm 3.



a) Pairwise Correspondences b) Initial Map for the Tile

Figure 4: Visualisation of the initial map creation for a single tile (hive location) using images from four different honeybee comb scans (captured at different time). First correspondences are established between pairs of images from individual scans (panel a), as described in Section 4.3.3. Then the initial map is calculated. Panel b) shows the initial map of the single tile overlayed over one of the original images.



a) Initial Maps for Individual Tiles b) Final Initial Map

Figure 5: Visualisation of the process of combining per-tile initial maps into the complete initial map of the comb. For simplicity, we show only four image tiles. Panel a) shows the initial maps of individual tiles, panel b) the finalised initial map (black) alongside all cell detections used for the estimation of the cell positions in the map (red).

Algorithm 2: Per-tile map creation for a tile location k from the first n comb scans

Input: *Images* $(\mathbf{I}_{s,k})_{s=1}^{n_{init}}$ *taken from location* k
Output: *Per-tile map* $P_k = \{c_{k,1}, \ldots, c_{k,N_k}\}$
$D \leftarrow \emptyset$;
`// Detect cells in images`
foreach *scan* $s = 1$ ***to*** n_{init} **do**
  $\mathcal{D}_{s,k} \leftarrow$ detect open cells in $\mathbf{I}_{s,k}$;
  $D \leftarrow D \cup \mathcal{D}_{s,k}$;
end
$T \leftarrow \emptyset$;
`// Estimate pairwise translations`
foreach $i, j \in \{1, \ldots, n_{init}\}$, $i \neq j$ **do**
  $C_{i,j} \leftarrow \text{register}(\mathcal{D}_{i,k}, \mathcal{D}_{j,k})$;
  $\mathbf{T}_{i,j} \leftarrow \frac{1}{|C_{i,j}|} \sum_{(\mathbf{d},\mathbf{d}') \in C_{i,j}} (\mathbf{d} - \mathbf{d}')$;
  $T \leftarrow T \cup \{\mathbf{T}_{i,j}\}$;
end
`// Refine image positions`
$\{\mathbf{p}'_{s,k}\} \leftarrow$ Least-Squares Optimization$(\mathbf{p}_{1,k}, T)$;
$C \leftarrow \emptyset$;
`// Cluster detections`
foreach *detection* $\mathbf{d} \in D$ *with position* $\mathbf{p}'_{\mathbf{d}}$ *and radius* $r_{\mathbf{d}}$ **do**
  if $\exists\, c_\ell = (\mathbf{p}_{c_\ell}, r_{c_\ell}, o_{c_\ell}) \in C : \|\mathbf{p}'_{\mathbf{d}} - \mathbf{p}_{c_\ell}\| < \frac{G}{2}$ **then**
    $o_{c_\ell} \leftarrow o_{c_\ell} + 1$;
    $\mathbf{p}_{c_\ell} \leftarrow \mathbf{p}_{c_\ell} + \frac{1}{o_{c_\ell}} \left(\mathbf{p}'_{\mathbf{d}} - \mathbf{p}_{c_\ell}\right)$;
    $r_{c_\ell} \leftarrow r_{c_\ell} + \frac{1}{o_{c_\ell}} (r_{\mathbf{d}} - r_{c_\ell})$;
  else
    $C \leftarrow C \cup \{(\mathbf{p}'_{\mathbf{d}}, r_{\mathbf{d}}, 1)\}$;
  end
end
$P_k \leftarrow$ Non-Maximum Suppression$(C, \text{IoU} > 0.3)$;

4.3.4. Updating Map

Once the initial map M_{20} is created, the subsequent comb scans $(S_s)_{s=21}^{n}$ are used to update it. For each new image $\mathbf{I}_{s,k}$ with $k \in \{1, \ldots, 462\}$ in the scan S_s, we apply cell detection-based registration (Alg. 1) to establish correspondences between the detected cells in the image and those already present in the map (see Fig. 6). Based on new observations, we update existing map cells, and detections without correspondences are incorporated into the map as new cells. To ensure spatial consistency, we again apply NMS. Algorithm 4 provides the pseudocode for the map update. When discarding cells from the map, we also need to construct the mapping matrix $A_{s-1,s}$ (see Section 4.1) between the cells in the map M_{s-1} and M_s, which allows for retrospective association of the corresponding cells in the map sequence.

Algorithm 3: Combining per-tile maps into a global initial map by matching and clustering detections

Input: *Set of per-tile maps* $P = \{P_1, \ldots, P_{462}\}$
Output: *Initial map* $M_{n_{init}} = \{c_{n_{init},1}, \ldots, c_{n_{init},N_n}\}$
$C \leftarrow \emptyset$;
`// Group matching detections`
foreach *neighboring per-tile maps* (P_i, P_j) **do**
  $C_{i,j} \leftarrow$ register(P_i, P_j);
  foreach *corresponding pair cells* $(c'_i, c'_j) \in C_{i,j}$ **do**
    $d \leftarrow$ detections_of$(c'_i) \cup$ detections_of(c'_j);
    if $\exists\, c_\ell \in C$ *s.t.* $d \cap c_\ell \neq \emptyset$ **then**
      $c_\ell \leftarrow c_\ell \cup d$;
    else
      $C \leftarrow C \cup d$;
    end
  end
end
`// Add unmatched detections`
foreach *per-tile map* $P_i \in P$ **do**
  foreach *cell* $c' \in P_i$ *and* $c' \notin \bigcup_{c \in C} c$ **do**
    $C \leftarrow C \cup \{$detections_of$(c'_i)\}$;
  end
end
$\mathcal{K} \leftarrow \emptyset$;
`// Recompute cells from detections`
foreach *cluster* $c = \{\mathbf{d}_1, \ldots, \mathbf{d}_m\} \in C$ **do**
  $\mathbf{p}_c = \frac{1}{m} \sum_{i=1}^{m} \mathbf{p}_{\mathbf{d}_i}$;
  $r_c = \frac{1}{m} \sum_{i=1}^{m} r_{\mathbf{d}_i}$;
  $o_c = m$;
  $\mathcal{K} \leftarrow \mathcal{K} \cup \{\{\mathbf{p_c}, r_c, o_c\}\}$;
end
$M_{n_{\text{init}}} \leftarrow$ Non-Maximum Suppression$(\mathcal{K}, \text{IoU} > 0.3)$;

4.4. Semantic Information in the Map

The spatial mapping provides a sequence of observations for individual cells in the comb. As discussed in Section 2, prior works have classified the cells based on their content by image classification neural networks. Nevertheless, this approach requires a well-annotated dataset of cell images for training, which may be challenging to obtain. The robotic system that we use was optimised for honeybee queen tracking Ulrich et al. (2024). Therefore, details of cell contents are often not clearly visible to humans (see Fig. 7), making standard manual annotation protocols practically infeasible.

Fortunately, the development stages of honeybees are well described in the literature, providing a strong prior knowledge that can be encoded using a model of temporal evolution. We chose to model this evolution using a simplified Markov model, which allowed us to formulate

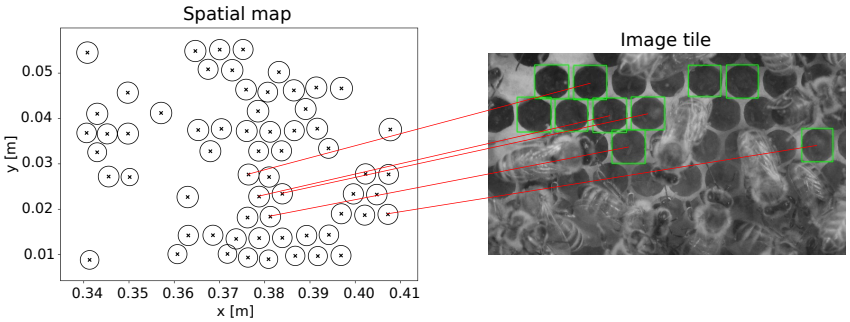


Figure 6: Visualization of registration of new observations into an existing spatial map. The new observation form the comb scan (right) is successfully matched with the map (left) by the correspondences (red). Observations that were not matched are cells which were missing in the map, because they were not observed before. By integrating this observation, position of the cells that were matched will be updated and new cells will be added into the map.

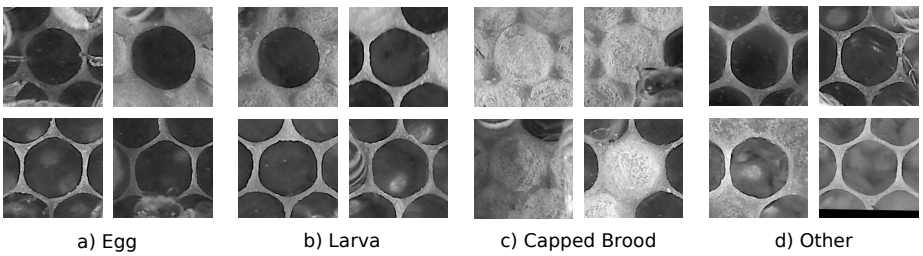


Figure 7: We consider four classes of cells based on their content: a) *egg*, b) *larva*, c) *capped brood* and d) *other* (storage or empty). Due to lighting, debris and other conditions inside of the hive, the cell content is typically not easily distinguishable by the human eye. This especially concerns eggs and young larvae. As shown later, a neural network-based algorithm was, however, able to learn to classify those to a certain extent.

Algorithm 4: Updating the comb map with a new scan

Input: *Map* $M_{s-1} = \{c_{s-1,1}, \ldots, c_{s-1,N_{s-1}}\}$, *scan* S_s
Output: *Updated map* M_s
$M_s \leftarrow M_{s-1}$;
foreach *tile* $k = 1$ ***to*** 462 **do**
    $\mathcal{D}_{s,k} \leftarrow$ detect open cells in $\mathbf{I}_{s,k}$;
    $C_{s,k} \leftarrow$ register($\mathcal{D}_{s,k}, M_s$);
    `// Update map cells`
    foreach *correspondence* $(c_i, \mathbf{d}) \in C_{s,k}$ **do**
        $o_i \leftarrow o_i + 1$;
        $\mathbf{p}_i \leftarrow \mathbf{p}_i + \frac{1}{o_i}(\mathbf{p}_\mathbf{d} - \mathbf{p}_i)$;
        $r_i \leftarrow r_i + \frac{1}{o_i}(r_\mathbf{d} - r_i)$;
    end
    `// Add new map cells`
    foreach *detection* $\mathbf{d} \in \mathcal{D}_{s,k} \setminus \{\mathbf{d} \mid (c_i, \mathbf{d}) \in C_{s,k}\}$ **do**
        $M_s \leftarrow M_s \cup \{(\mathbf{p}_\mathbf{d}, r_\mathbf{d}, o = 1)\}$;
    end
end
$M_s \leftarrow$ Non-Maximum Suppression(M_s, IoU > 0.3);

a Bayesian filter (see Section 4.4.1). By integrating the chain temporal model with multiple specialised sensors providing observations on the current hidden state, we can estimate the state of the cells over time. Our system employs three visual sensors, one for classifying capped brood cells (Section 4.4.3)—easily recognisable by a human annotator—and another for classifying open brood cells into either *other* or joint "open brood" categories (Section 4.4.2). Since the robot tracks the honeybee queen most of the time, the third sensor detects egg-laying events (Section 4.4.4), providing crucial information on positions and times of newly laid eggs.

4.4.1. Temporal Model of Honeybee Evolution

The temporal model we used is a discrete chain description of the biological process that occurs within the brood cells. We use it for a filtering task, i.e., to estimate the posterior distribution of belief over the true hidden state of the cell $H_t \sim p(H_t \mid z_{1:t})$ at time t for each cell in the map given its past observations $z_{1:t}$. In our work, we follow the formulation and notation of the Bayes filter from Thrun et al. (2005). The Bayes filter simplifies the filtering process by assuming Markov property, meaning that the current state encapsulates all relevant information for predicting future state. This can be expressed mathematically as $p(Z_t \mid H_t, z_{1:t}) = p(Z_t \mid H_t)$ and $p(H_t \mid H_{t-1}, z_{1:t-1}) = p(H_t \mid H_{t-1})$. The Markov assumption allows us to recursively calculate the posterior distribution $p(H_t \mid z_{1:t})$ at time t using the corresponding posterior $p(H_t \mid z_{1:t-1})$ at time $t-1$ and the most recent measurement z_t. The algorithm comprises two steps—prediction $p(H_t \mid z_{1:t-1})$ and measurement update $p(H_t \mid z_{1:t})$. The calculation follows

$$p(H_t \mid z_{1:t-1}) = \sum_{h_{t-1}} p(H_t \mid H_{t-1} = h_{t-1}) \cdot p(H_{t-1} = h_{t-1} \mid z_{1:t-1}) \tag{2}$$

$$p(H_t \mid z_{1:t}) \propto p(z_t \mid H_t) \cdot p(H_t \mid z_{1:t-1}), \tag{3}$$

where $p(H_t \mid H_{t-1})$ is referred to as the transition model and $p(Z_t \mid H_t)$ as the sensor model. In our work, we actually integrate three different sensors, with each providing different information for the state estimation, see Sections 4.4.2–4.4.4.

States of the Temporal Model. We distinguish four classes of cells—*egg*, *larva*, *capped brood* and *other*. Each developmental stage of a young bee has its duration, typically expressed in days Gullan and Cranston (2014). In order to not violate the Markov assumption of the Bayes filter, we define a separate state for each day within a developmental stage. Although we experimented with a finer time resolution of the temporal model (12-hour, 6-hour steps), we observed no significant improvement. Additionally, because the developmental stages of larvae and capped brood differ for female worker bees and male drones, we introduce two alternative paths for those states, which can be distinguished from the observations based on their duration. We do not model rare queen cells in the temporal model, as their appearance differs, and no swarming occurred during data collection. In total, the model consists of 43 states (see Table 2), as visualized in Fig. A.17 in Appendix A.1.

Content Class	Sex	Number of States
other	-	1
egg	-	3
larva	worker	5
	drone	7
capped brood	worker	13
	drone	14

Table 2: Number of states of the temporal model per each defined content class. For the cell states *egg*, *larva* and *capped brood* this corresponds to the duration of the developmental stage in days.

Transition Model. The transition model defines how individual cell states evolve over time. The illustration of the temporal model is provided in Fig. A.17 in Appendix A.1. As our work is exploratory in the use of long-term cell state tracking, the amount of information in literature to inform transition model design is limited. The transition probabilities may also vary depending on the specific hive, current weather or food storage. Therefore, we set the probabilities based on reasonable assumptions while acknowledging that the true probabilities may differ. Any unspecified transition probabilities are computed as the complement to the specified ones. The final *egg* state transitions to the first *larva* states, with a probability proportional to the worker-to-drone ratio in the hive, which we set to a sensible value of 20:1. Similarly, the last *larva* states for both worker bees and drones transition to their respective first capped brood states. Within each developmental stage, states are connected sequentially over time. Additionally, each state can revert to *other*—due to the possibility of brood cannibalism—with probabilities $p(\textit{egg}\text{ can.}) = 0.05$, $p(\textit{larva}\text{ can.}) = 0.01$ and $p(\textit{capped brood}\text{ can.}) = 0.001$. For the first egg state, this connection is bidirectional to account for the possibility that with probability $p(\textit{egg}\text{ emergence}) = 0.05$, the queen may lay an egg in an empty cell. When setting the probabilities of cannibalism for individual states, we need to address that the probability of the removal at a certain age is conditioned on brood surviving to that age. We will now briefly describe the calculation of the probabilities of brood cannibalism for each day of the class k development stage, where k is either *egg*, *larva*, or *capped brood*. The probability $p(k\text{ can.})$ for state of class k that forms N days long sequence

of states $(k_i)_{i=1}^{N}$ can be expressed as:

$$p(k \text{ can.}) = \sum_{i=1}^{\mathrm{N}} P(k \text{ can.} \mid k_i) P(k_i) \tag{4}$$
$$= \sum_{i=1}^{\mathrm{N}} P(k \text{ can.} \mid k_i) \prod_{j<i} (1 - P(k \text{ can.} \mid k_j)),$$

i.e., the total probability of cannibalism for a given stage k is the sum of probabilities for each day i given that the cannibalism did not occur before. We adopt a simplifying assumption that the probability of brood cannibalism is the same for each day of the development stage (it is equally likely on day 1 as day 2). Using this assumption, we further simplify the equation, which allows us to numerically solve it and set the probabilities $p(k \text{ can.} \mid k_{day})$ accordingly:

$$p(k \text{ can.} \mid k_i) = c \quad \forall i \in \{1, \ldots, N\} \tag{5}$$
$$p(k \text{ can.}) = \sum_{i=1}^{N} c \cdot (1 - c)^{i-1}.$$

Integrating Observations. The temporal model is designed for a one-day frequency of observations. However, the observations of the comb are not regular and usually do not come once a day, as the comb scans are performed when the honeybee queen is inactive, and the individual cells may not always be observable due to their occlusions by bees. To address that, if the elapsed time between the observation and the last prediction step is less than one day, only the update step (Eq. 3) is applied. Similarly, if the interval spans multiple days, the prediction step (Eq. 2) is performed multiple times before the update step. Before integrating the first observation of a map cell, its state is initialized with a uniform probability distribution. When integrating the observations, we add 5% uncertainty to the current state for numerical stability.

4.4.2. Sensor B: Open Brood Classification

In order for the temporal model to navigate through the possible states, we need to use so-called sensors that provide observations about the state of the cells. These observations are incorporated as new evidence to the probabilistic model, updating the belief about the cell state. The first such sensor is an open brood classifier, which processes images of detected cells, resizes them to 128×128 px, and classifies them into two categories *other* and joint "open brood" (encompassing both *egg* and *larva*). For this binary image classification task, we use a convolutional neural network (CNN) ResNet-9 trained on our custom dataset. Details on the annotation process and the resulting dataset are provided in Section 5.4, while the construction of the sensor model and its parameters is described in Section 5.5.1.

4.4.3. Sensor C: Capped Brood Classification

The temporal model inherently introduces a delay in state estimation, which may, for example, result in predictions of the *larva* state even when the cell has been already capped. To address this, we leverage images of cells that were not detected in the comb images—either due to occlusion or because they were already capped. To get images of cells that were not detected, we look at the positions where we expect them to be based on current information in the map. We trained the ResNet-9 CNN for binary image classification, processing 128×128 px cell images

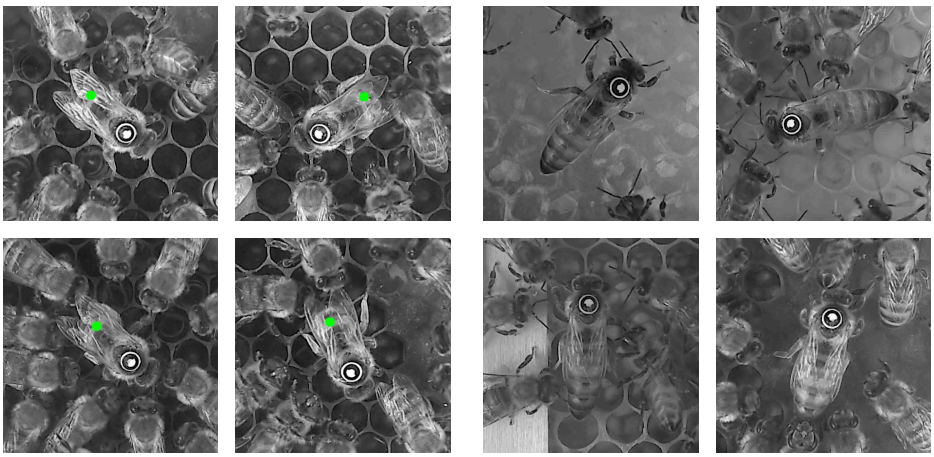


Figure 8: Candidate egg-laying events are either confirmed or rejected by the neural network. In a) are examples of the frames closest to the candidate timestamp for confirmed egg-laying events, together with the estimated cell position (in green), and b) shows false candidate events.

and classifying them as either *capped brood* or a joint "otherwise" class (encompassing undetected occluded open cells and capped honey cells). Details on the annotation process and used dataset are provided in Section 5.4, while the sensor model is described in Section 5.5.2.

4.4.4. Sensor E: Honeybee Queen Egg-Laying Detection

Egg-Laying Detection. Honeybee eggs are typically not visible in cell images, so we instead detect egg-laying events based on the queen's movement pattern. We adopt a similar pipeline as in our previous work Ulrich et al. (2024). First, we identify candidate egg-laying events by identifying when the queen stops moving. Next, we extract 3-second video segments centred on the candidate egg-laying timestamp. To classify these events, we train a convolutional neural network (ResNet-9) on a manually annotated dataset. During training phase, in each epoch, we randomly sample a single frame from all video segments in the training dataset. For inference, we classify each frame of the video and compute the average prediction to determine the final classification. Before passing the video frames to the neural network, we rotate the images in a way that the queen always faces upwards. Examples of the frames closest to the candidate timestamp both for egg-laying events and false candidate events are shown in Fig. 8.

Egg-laying Positioning. Beyond detecting egg-laying events, it is crucial to determine the exact cell of the map in which the eggs were laid. Given the potential noise in the queen's position and orientation, we employ a voting-based approach to improve robustness. For each frame in the 3-second video segment, we accumulate votes based on the estimated position and orientation of the queen on the comb. The cell with the highest number of votes is selected as the egg-laying location, with confidence proportional to the vote count. This method assumes that the spatial map is already well-initialized, but this is a reasonable assumption for the long-term deployment of such a mapping system. The details about the dataset and construction of the egg-laying sensor model are provided later in Section 5.5.3.

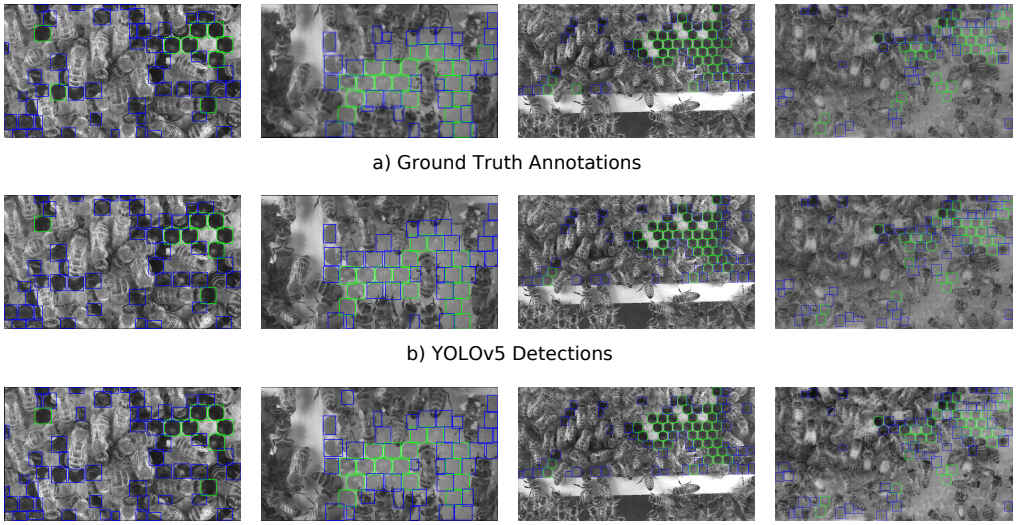


Figure 9: Samples from the test part of the cell detection dataset. Fully visible cells are highlighted in green, while partially occluded cells are in blue. a) shows the ground truth annotations, b) shows the detections provided by the YOLOv5, and c) shows the detections provided by the Faster R-CNN with ResNet-50 backbone.

5. Experiments and Results

In this section, we present the evaluation of the individual components of the mapping pipeline. First, we assess the performance of cell detection methods (Section 5.1). Next, we evaluate the quality of spatial registration in Section 5.2. We analyse the performance of the egg-laying detection model (Section 5.3) and the cell image classification networks (Section 5.4). Subsequently, we describe the construction of the sensor models for the temporal filter (Section 5.5). Finally, in Section 5.6, we evaluate the cell state estimation and in Section 5.7, we demonstrate some practical applications of our work.

5.1. Honeybee Cell Detection

For training and evaluation, we use an extended version of the cell detection dataset described in our previous work Janota et al. (2024). Both Faster R-CNN and YOLOv5 models were pre-trained on the COCO dataset Lin et al. (2014). The dataset comprises randomly selected images across different comb and time, with annotated bounding boxes for each image. As mentioned in 4.3.1, we distinguish between two cell categories: *fully visible* and *partially occluded* cells.

The dataset extension annotation followed the same semi-automated process as in Janota et al. (2024). We first applied the Segment Anything Model (SAM) by Kirillov et al. (2023) with the ViT-H model for instance segmentation, filtered the resulting masks based on area and circularity, and manually reviewed and corrected the annotations. The final dataset contains 497 images with resolutions 67 µm and 37 µm per pixel, split into training (363 images), validation (61 images) and test (73 images) subsets. Fig. 9 shows sample images from the test set, together with the cell detections from the best trained models.

Method	AP [%]	AP-50 [%]	P [%]	R [%]	F1-score [%]
CHT (Bilateral f., CLAHE)	9.4	13.4	11.2	89.9	19.9
YOLOv5s6	87.6	93.5	**93.2**	90.5	**91.8**
Faster R-CNN (ResNet-18)	81.1	92.5	87.2	89.9	88.5
Faster R-CNN (ResNet-50)	**91.3**	**95.3**	90.3	**93.3**	**91.8**

Table 3: Cell detection results for the class of *fully visible* open cells.

Method	AP [%]	AP-50 [%]	P [%]	R [%]	F1-score [%]
CHT (Bilateral f., CLAHE)	-	-	-	-	-
YOLOv5s6	71.6	**89.8**	**84.1**	83.7	83.9
Faster R-CNN (ResNet-18)	58.8	82.9	80.4	76.9	78.6
Faster R-CNN (ResNet-50)	**75.2**	88.9	82.1	**86.2**	**84.1**

Table 4: Cell detection results for the class of *partially occluded* open cells.

The models were evaluated on the test part of our dataset. We applied class-agnostic NMS with an IoU threshold of 0.3 on the output of the object detectors (where applicable). Detections from the CHT are considered to be of class *fully visible* cell.

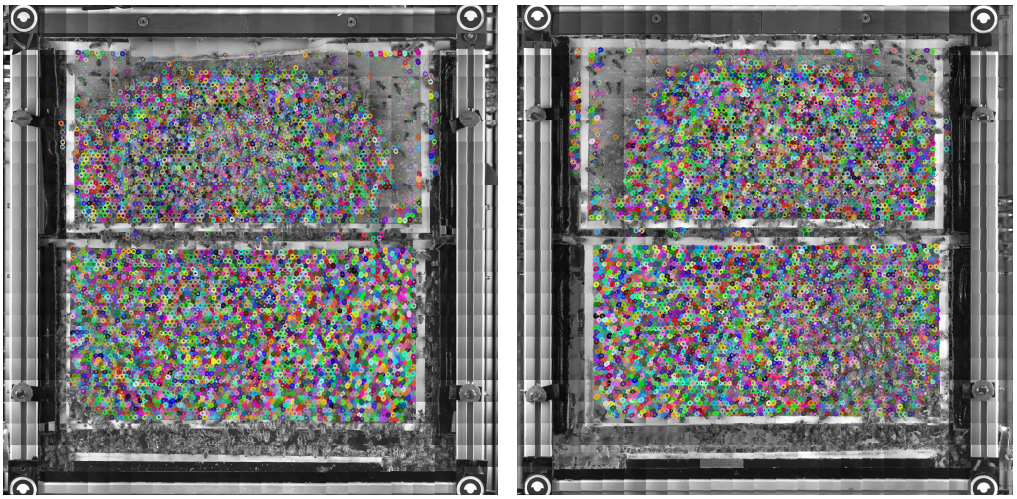
For each class, we computed the Average Precision (AP) metric at an IoU threshold of 0.5, as well as the average AP over IoU thresholds in the range [0.5,0.95] with a step size of 0.05. Additionally, we report precision (P), recall (R) and F1-score at an IoU threshold of 0.5 and a confidence threshold of 0.5. The results for the class *fully visible* cell are summarized in Table 3 and for the *partially occluded* cell are presented in Table 4.

Although CHT achieved high recall, it generated many false positives, making it unsuitable for images containing bees without additional filtering. The best performance was achieved by the Faster R-CNN with a ResNet-50 backbone. YOLOv5 achieved comparable results, as also illustrated in Fig. 9. Faster R-CNN demonstrated higher recall but lower precision across both cell classes, whereas YOLOv5 showed lower recall but higher precision. Based on observations from the test dataset, Faster R-CNN detects even poorly visible open cells. However, this sensitivity also leads to more false positives, as it sometimes incorrectly identifies capped brood and honey cells with distinct borders as open cells. In contrast, YOLOv5 produces fewer false positives but is more likely to miss partially visible cells and, somewhat more frequently than Faster R-CNN, to misclassify fully visible cells as partially occluded. While YOLOv5 performed slightly worse, we selected it for our mapping pipeline instead of the Faster R-CNN due to its lightweight architecture and higher computational efficiency.

5.2. Spatial Registration

The ground truth for the spatial map is practically infeasible to create, as it would require annotating correspondences between all possible pairs of images in individual comb scans and between adjacent comb scans. Instead, we employ two alternative performance indicators.

To evaluate the precision of odometry and spatial registration, we compute the distribution of distances between the individual cell detection positions and their corresponding map cell positions. Due to the spatial registration, the distances cannot be larger than half the size of the cell, however, we can infer about the precision of the odometry from the tail of the distribution. In an ideal scenario, the tail should be short and light, meaning most detections align closely with their expected positions. However, in cases of poor odometry, the tail extends further, indicating



a) Side 0 b) Side 1

Figure 10: Visualization of all cell detections on both sides of the comb. The cell detections that were assigned to the same cell cluster are highlighted with the same colors, showing that the matching of detection leads to a spatially consistent map.

significant misalignments and larger deviations in the estimated positions. The distributions for both sides of the hive are shown in Fig. A.18 in Appendix A.2. It can be seen that the majority of the distances are concentrated well bellow the required threshold.

Secondly, we provide a qualitative performance indicator by visualizing all cell detections and highlighting them with colors based on their corresponding map cell. The images of the entire comb with visualized cell detections are shown in Fig. 10. The well-defined cell clusters in the images, again, suggest that the odometry and spatial registration are sufficiently precise for our application.

After generating the initial map from the first 20 full comb scans, we have observed about 80% of all the cells detected by the end of our dataset. The number of observed cells then increased gradually. It should be noted that the final map did not cover certain areas of the comb with capped honey (see Fig. 14) because capped honey cells could only be added to the map if they had been previously observed in their open stage.

5.3. Honeybee Queen Egg-laying Detection

As mentioned in Section 4.4.4, the egg-laying detection component classifies 3-second video segments in cases when the queen stops moving. These segments were annotated as either egg-laying events or false candidate egg-laying events. For training and evaluation of the binary classification network, we manually annotated a total of 1075 candidate egg-laying events, which were divided into training (698 video segments: 365 negative, 333 positive samples), validation (184 video segments: 92 negative, 92 positive samples), and test (193 video segments: 98 negative, 95 positive samples) subsets.

The performance of the ResNet-9 classification network is measured using standard classification metrics: precision, recall and F1-score, calculated on the test dataset. The results are presented in Table 5, the egg-laying detector achieved near-perfect results. However, it should

be noted that the recall and precision of the egg-laying detection may be lower in the actually deployment. This is because the honeybee queen may not be always observed—tracking failures or comb scanning can lead to missed honeybee queen activity. Additionally, we observed instances where an egg-laying event was correctly detected, but the queen most likely did not deposit any egg in the cell.

5.4. Honeybee Cell Image Classification

We trained two distinct classifiers: one that classifies open cells into *other* and "open brood" classes (see Section 4.4.2) and another that classifies undetected cells into *capped brood* and joint "otherwise" classes (see Section 4.4.3). For training and evaluation we manually annotated randomly selected cell sequences from the spatial map.

Annotating open-cell images is challenging for human annotators, as the content of the open cells is usually not clearly visible (see Fig. 7). However, capped brood cells are easily identifiable, and thus, serve as the starting point for our annotation process. To extract them from the comb scans, we look where the map expects currently not detected cells—either open cells occluded by bees or capped cells. As the first step, we manually annotated all *capped brood* cells among those images. Using knowledge of the developmental stage durations, we then retrospectively annotated open *egg* and *larva* cells, as each capped cell must have passed through these stages. Figure 11 illustrates the annotation process. If an egg-laying event was detected prior to the capping, we annotated all cells from the time of egg-laying until capping. Otherwise, we annotated cell observations starting eight days before capping and excluded those from ten to nine days before capping, as it also might have been a drone whose open brood stage is two days longer. In cases when it was not clear whether the state should be *egg* or *larva*, for example, during the transition from one to another, we annotated it as a joint "any open brood" class. Additionally, for detected egg-laying events that did not result in capping, we annotated the following two days as *egg* and excluded from the dataset cell images observed on the third day after egg-laying. All other open cell observations were annotated to be of class *other*.

In total, we annotated 531 cell image sequences, which were split into training (284 sequences), validation (79 sequences) and test (168 sequences) datasets. For the total number of images in each considered class, see Tab. A.10 in Appendix A.3. During the training of the open brood classifier, we use the detected open cells in the sequences, on the other hand, for training the capped brood classifier, we use the cells that were not detected but extracted from the images at their expected positions. Due to class imbalance, particularly the high number of *other* cells in the open cell classifier dataset, we downsampled this class and used only one-tenth of these samples during training of the open brood classifier to mitigate overfitting. Similarly, for the capped brood classifier, we reduced the number of non-capped brood samples to one-third during training.

The models were evaluated on the test dataset using standard classification metrics: precision, recall and F1-score. The results are presented in Table 5. The capped brood classifier performed well, achieving an F1-score of 91.4%. In contrast, the open brood classifier showed lower performance, with an F1-score of 77.4%. This result is expected, as the contents of open brood cells are often difficult to distinguish, even for human observers (see Fig. 7).

5.5. Sensor Models for the Temporal Filter

To integrate observations into the temporal model, we must estimate the sensor model, defined as the probability $p(Z_t \mid H_t)$, see Section 4.4.1. In this section, we go through the settings of sensor models for the open brood classifier (Section 5.5.1), capped brood classifier (Section 5.5.2)

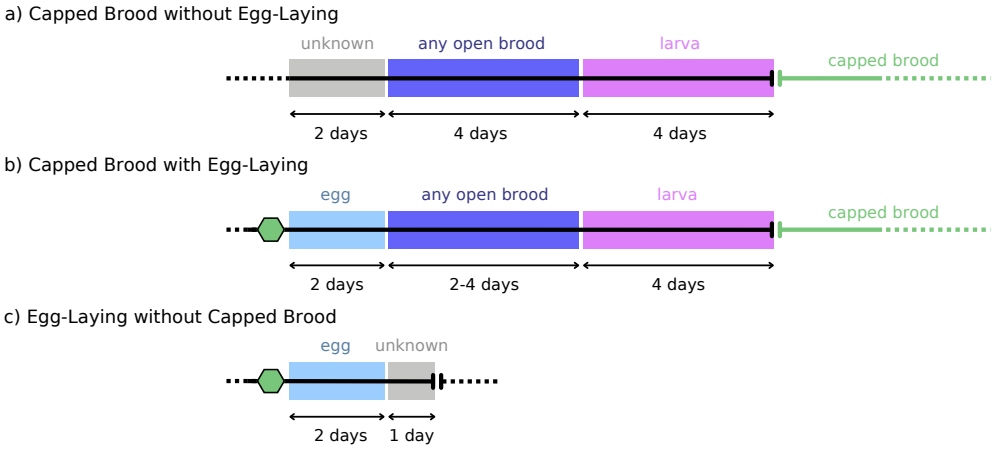


Figure 11: Visualization of the annotation process of detected open cells: (a) when no egg-laying was detected before capping, (b) when an egg-laying event occurred before capping, and (c) when an egg-laying event was observed but was not followed by capping. Using the knowledge of developmental stage durations, we can infer annotations even when the cell content is not clearly visible to human annotators.

Classifier	**Precision** [%]	**Recall** [%]	**F1-score** [%]
Egg-laying	100.0	98.9	99.4
Open brood	77.3	77.6	77.4
Capped brood	91.1	91.8	91.4

Table 5: Classification metrics for each ResNet-9 binary classifier.

Observation Type	*Egg*	*Larva*	*Capped Brood*	*Other*
Open brood	0.848	0.848	(0.001)	0.257
Other	0.152	0.152	(0.001)	0.743
No observation	(0)	(0)	(0.998)	(0)

Table 6: Parameters of the open brood (B) sensor model. For classes with multiple states, the parameters are shared.

Observation Type	*Egg*	*Larva*	*Capped Brood*	*Other*
Capped brood	0.166	0.166	0.872	0.166
Otherwise	0.834	0.834	0.127	0.834

Table 7: Parameters of the capped brood cell (C) sensor model. For classes with multiple states, the parameters are shared.

and egg-laying detector (Section 5.5.3). While for some sensor model parameters, we were able to get reasonable estimates using data—especially from classifier validation data—in some cases, we had to resolve to expert judgment because necessary data was not available.

5.5.1. Sensor Model B: Open Brood Classification

For the parameters of the open brood classification sensor model $p(Z_t \mid H_t)$, we use a properly normalized confusion matrix calculated using the validation dataset, described in Section 5.4. The values are duplicated for *egg* and *larva* state as the sensor does not differentiate them, values for *capped brood* have been set experimentally. Based on temporal model evaluation on validation data, we incorporated 20% of uniformly distributed noise into the confusion matrix. An important part of the sensor model is then the introduction of a virtual observation "no observation". Practically, open cell detection can only happen when the cell is open (either *egg*, *larva* or *other*), not when the state is *capped brood*. The observation is virtual because it never actually happens in the system, so it is never integrated, it simply allows to model that observing an open cell is actually evidence of it not being closed. Technically this also results in proper normalization. Thus, for the capped brood states, we set the probability of open cell observation $p(Z_t \mid H_t = \text{capped brood}) = 0.001$. The final sensor model parameters are summarized in Table 6 with values estimated without data in brackets.

5.5.2. Sensor Model C: Capped Brood Classification

The capped brood sensor model $p(Z_t \mid H_t)$ is computed following the approach outlined in previous Section 5.5.1. Specifically, the capped brood classification model is evaluated on a validation dataset (see Sec. 5.4), and the normalized confusion matrix serves as the sensor model once we duplicate states *egg*, *larva* and *other* that are aggregated on the classifier output. Additionally, based on the evaluation of the temporal model on validation data, we add 20% uniformly distributed noise to the sensor model. The sensor model parameters are shown in Table 7. The capped brood sensor is integrated only when one of the following holds: (1) the current state estimate of the cell is *larva*, and the capped brood classification confidence exceeds 0.5, or (2) the current state estimate is not *capped brood* and the capped brood classification confidence exceeds 0.8.

5.5.3. Sensor Model E: Egg-laying detection

The egg-laying sensor model should account for both the accuracy of candidate event classification and the precision of associating the position of the queen with the correct cell in the map. To derive the sensor model, we manually annotated 390 randomly selected candidate egg-laying events, evaluating whether both the classification and positioning were correct. The sensor model for *egg* state in day 1 is calculated using the results (212 candidates were correctly classified as egg-laying events and positioned, 6 candidates were correctly classified as egg-laying events but incorrectly positioned, 2 candidates were misclassified as non-egg-laying events, 170 candidate events were correctly classified as non-egg-laying).

We know that a queen inserting her abdomen into a cell can only happen over an empty cell (and we also assume that in such case, she always lays an egg). Now, using the above data, we can compute the ratio

$$\frac{p(Z_t = \text{egg-laying, correct position} \mid H_t = \text{egg}_0)}{p(Z_t = \text{egg-laying, incorrect position} \mid H_t = \text{egg}_0)}. \tag{6}$$

However, to get the probabilities, we also need to know how often observation "no egg-laying" happens when the cell is in state egg_0. That means knowing how often she shortly stops over cells in state egg_0 relative to all other states. Unfortunately, we cannot determine from the images over which type of cell it happened, so we have to make an assumption to calculate the parameters of the sensor model.

We chose to assume the ratio $|\text{no egg-laying} \wedge \text{egg}_0|/|\text{no egg-laying}|$ to be about one fourth, which is likely exaggerated but we experimentally validated the results are not sensitive to it. This leads to the probabilities in Table 8. Notably, we do omit the performance of candidate selection, which depends on factors such as potential missed observations due to scanning and occasional failures in queen detection.

The probabilities of observing an egg-laying event for all other states are arbitrarily set to 0.001, reflecting that the physical egg-laying behaviour cannot happen over a closed or full cell, so such an observation would only happen due to error. The egg-laying sensor is used only for positive egg-laying events—with the confidence of the classification neural network greater than 0.5—because if the candidate event is a false one, it does not provide any additional information to the temporal model. The "no egg-laying" observation is, thus, similar to the case in Section 5.5.1, a virtual observation, which is never integrated into the temporal model. Additionally, when integrating the observations, we use position confidence to weigh the probabilities of "egg-laying, correct position" and "egg-laying, incorrect position". The parameters of the sensor model can be seen in Table 8 with those which were not estimated from the data denoted in brackets.

An important detail is the clock synchronization, we have to make sure that the day step of the model corresponds to the development of the bee. When the egg-laying observation is integrated, we reset the timestamp of the filter to synchronise subsequent filtering with the egg-laying event. This is an engineering choice to deal with the fact that the *other* class does not follow any specific temporal pattern in the daily rhythm of the model.

5.6. Honeybee Cell State Estimation

The egg-laying detector and image classification networks do not directly provide an estimate of the cell state. To address this, we employ a Bayes filter that integrates observations from these components to estimate the cell state over time. In this section, we first conduct an ablation

Observation Type	*Egg* Day 1	Other Classes
Egg-laying, correct position	0.808	(0.001)
Egg-laying, incorrect position	0.023	(0.001)
No egg-laying	0.169	(0.998)

Table 8: Parameters of the egg-laying (E) sensor model.

study to demonstrate the importance of utilizing all the proposed egg-laying detection and image classification modules. Next, we present per-class classification metrics to evaluate the performance of the temporal model. We then showcase the temporal model's ability to detect capping events and its capability to predict capping events several days in advance in the absence of new observations. For all evaluations in this section, we use the dataset described in Section 5.4.

5.6.1. Ablation Study of Sensors for the Temporal Model

We present an ablation study evaluating the contribution of different sensors in the temporal model (see Fig. 12). The results demonstrate that each sensor plays a crucial role in achieving accurate cell state estimation. The egg-laying sensor significantly enhances the classification of the *egg* state, which as can be seen, the open cell image classification network alone struggles with, as eggs are often not visible in images. When combined, the egg-laying and open brood classification sensors already yield satisfactory performance for most classes. However, due to natural variations in brood development, distinct development durations for worker bees and drones, and the delayed nature of the temporal model, some capped brood cells may initially be misclassified as larva. Incorporating the capped brood visual sensor further refines the classification and improves the detection of the capping events early. We also statistically compare ECB to the second-best EB using a permutation test on the reported per-class F1-scores, where randomisation of labels is used to simulate the null hypothesis of equal performance. EBC significantly outperformed EB for *other* (Δ F1-score = 0.026, $p < 0.001$, n = 8963), *larva* (Δ F1-score = 0.341, $p < 0.001$, n = 599), and *brood* (Δ F1-score = 0.128, $p < 0.001$, n = 6039), while the improvement for *egg* (Δ F1-score = 0.024, p = 0.194, n = 192) was not statistically significant due to small sample size. Therefore, we employ the temporal model with all three sensors in the subsequent experiments.

5.6.2. Classification Metrics

Since the temporal model provides a comprehensive estimate of the cell state, we evaluate its performance across all considered cell classes: *egg*, *larva*, *capped brood*, and *other*. For each annotated cell sequence, we initialize the state with a uniform distribution and iteratively update it based on observations. If an observation is annotated, we compare it against the estimated state. To assess performance, we compute standard classification metrics—precision, recall, and F1-score—separately for each class.

The results, summarized in Table 9, indicate that the temporal model achieves near-perfect classification for the *other* and *capped brood* classes. The classification of *larva* cells is also highly reliable, though slightly worse. The most challenging class to classify is *egg*. This is likely due to two main factors: (1) eggs are often not clearly visible in images, making detection inherently difficult, and (2) during observed egg-laying events, the honeybee queen may not always deposit an egg in the cell. Overall, these results demonstrate the effectiveness of the temporal model in integrating multiple sensor observations to achieve cell state estimation across all the developmental stages of the honeybees, even when the state is not directly observable.

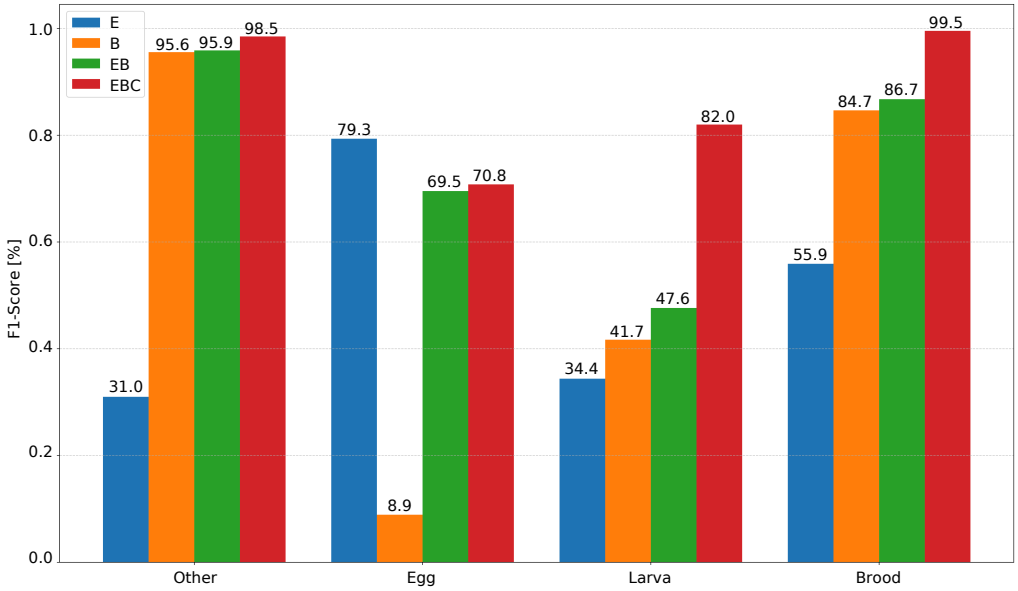


Figure 12: Results of the ablation study comparing the overall system performance using different sensor combinations (E—egg-laying sensor, B—open brood sensor, EB— a combination of E and B, EBC— a combination of E, B and sensor of capped brood). Per each class, we show F1-scores of the filtered classifications. Since each sensor contributes to the classification of a certain class we see that the combination of all EBC (red) performs the best overall.

Class	**Precision** [%]	**Recall** [%]	**F1-score** [%]
Other	98.5	98.5	98.5
Egg	72.9	68.8	70.8
Larva	77.7	86.8	82.0
Capped brood	100.0	99.1	99.5

Table 9: Filtered classification performance on the test dataset using all sensors.

5.6.3. Delay of Temporal Filter on Capping Events

Beyond standard classification metrics, we are also interested in the inherent delay that is induced by the temporal filter. Because the capping of brood cells constitutes a natural breaking point, which we can easily and reliably annotate, we chose to test how precisely in time the temporal model correctly identifies the capping events. Since observations occur at irregular time intervals, we generate a sequence of state estimates with a fixed one-day time step. To achieve this, we start with the first available observation for each individual cell and sequentially integrate subsequent observations using the temporal model, continuously refining the state estimate. Rather than storing estimates at every update, we only save state estimates at fixed one-day intervals. Whenever the next scheduled one-day time step is reached, we take the current state estimate, perform a prediction step using the temporal model, save the predicted state, and proceed with integrating the following observations. This ensures that the final sequence of saved state estimates maintains a uniform one-day step while incorporating all available observations.

For evaluation, we compare the estimated state sequences to the ground truth observations of capped brood. The ground truth timestamps indicate the first detected occurrence of capped brood, though they may not precisely correspond to the actual capping event due to irregularity of observations. We then measure the time difference between the first estimated occurrence of capped brood in our generated sequence and the first observed occurrence in the ground truth sequence. A capping event is considered detected if this time difference is no more than three days.

In the test part of the dataset, the temporal filter correctly estimated 96.8 % of capping events (61 out of 63), with a mean time difference and standard deviation of (0.14 ± 0.78) days. This slight lag is expected, as any temporal filter is delayed in nature. Nevertheless, the discrepancy remains below the one-day discretization step, demonstrating the temporal model's reliability in accurate detection of capping brood events with minimal temporal offset.

5.6.4. Forecasting Capping Events

In an ideal scenario, the temporal model would be used to predict the future development of all comb cell states. However, achieving reliable predictions requires careful tuning of the model's parameters. The classification of *egg* and *larva* cells is particularly challenging (see Section 5.6), and accurately forecasting their future states is even more difficult. This is primarily due to variability in the egg-laying rate, which depends on the colony's current condition, making it difficult to predict the number of *egg* and *larva* cells with high confidence.

However, *capped brood* cells can be classified with high accuracy (see Section 5.6) and given the estimated brood cannibalism rate and the current observed number of *egg* and *larva* cells, it should possible to predict future capping events. In this section, we conduct an experiment to assess how many days into the future the temporal filter can accurately forecast capping events. This is complementary to experiments with the detection time accuracy of Section 5.6.3 as now we ask how long into the future capping can be accurately predicted, given that detecting it is actually equivalent to predicting it 0 days into the future.

For evaluation, we use the test dataset described in Section 5.6 and generate state estimates at uniform one-day intervals, as outlined in previous Section 5.6.3. To make predictions into the future, we incorporate an estimated brood cannibalism rate (see Section 5.7.1) into the temporal model by adjusting the probability of cannibalism for *egg* states and the first three days of the *larva* state.

For each ground-truth sequence of cell observations, we identify subsequences with capped brood and assess the predictive performance of the temporal filter. We simulate future state

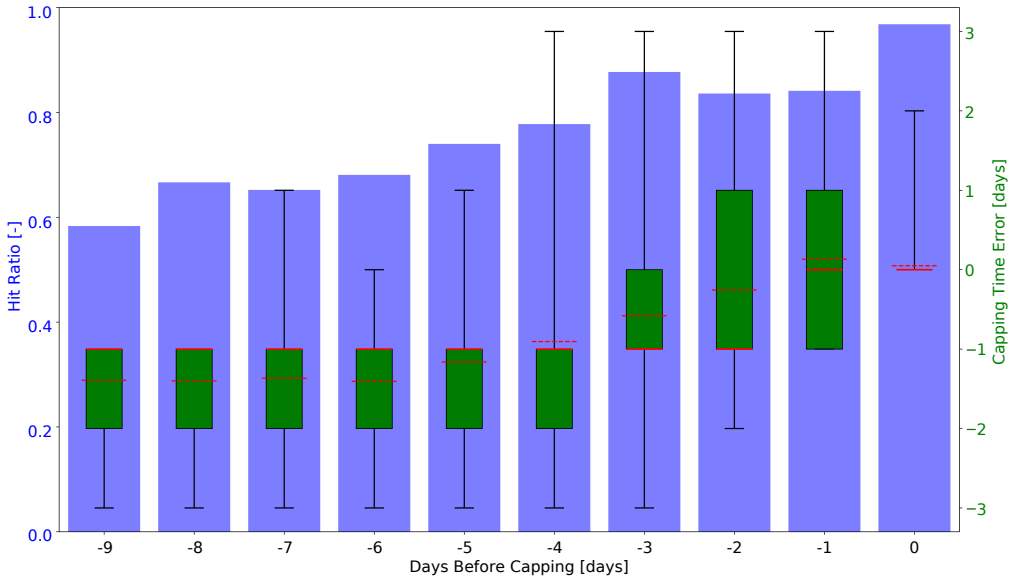


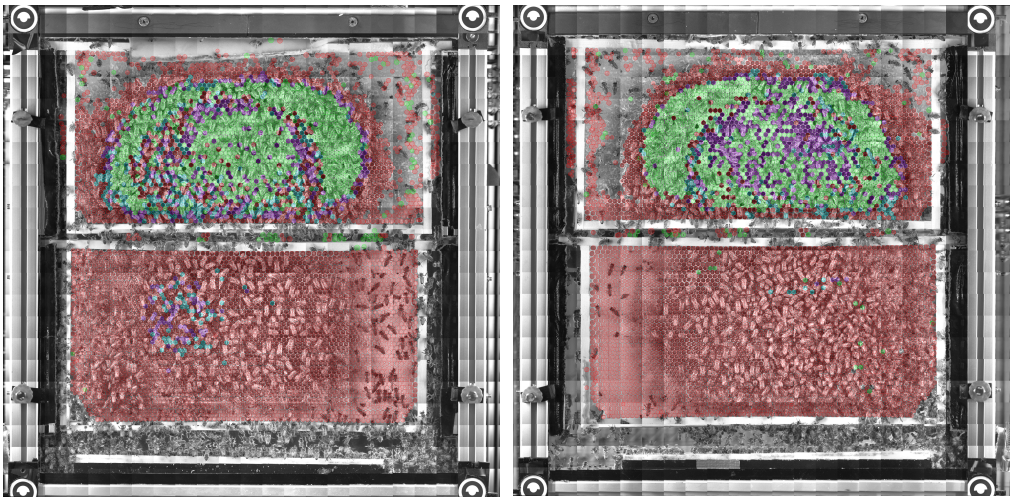
Figure 13: The performance of the system on the task of capping-time prediction given a number of days ahead up to day 0 which corresponds to the simple filtered state estimation. The blue barplot shows the proportion of capping events that were correctly predicted. As expected, the hit ratios grow as the capping event is approaching, up to 96.8 % of filtered classification hit rate in day 0. The boxplot shows the associated error in the predicted time of the capping event with with median (solid red) and mean (dashed red) values.

predictions without any additional observations at several points before the capping event starting nine days prior. Note that because the sequence of observations before the capping event might not be long enough, we need to normalize our metrics only over the set of those that are, e.g., an event on the third day cannot be predicted nine days prior. Prediction accuracy is measured using a hit ratio, defined as the proportion of capped brood sequences that were predicted that far into the future. A capping event is considered successfully predicted if the estimated time falls within ± three days of the actual ground-truth event, to account both for the difference in duration of worker and drone developmental stages and a possible time error of the temporal filter.

Figure 13 presents the hit ratios alongside the associated error in the predicted capping times. The results show that our system is able to predict capping events well into the future. Notably, six days before the capping event, the model successfully predicts almost 70% of capped brood sequences. To ensure consistency in error evaluation, we round the capping time errors towards zero, meaning that predictions made on the same day—even with some small delay—are considered to have no error. Until one day before capping, the temporal model generally predicts the event on average a day too early. However, from that point onward, the inner time clock of the temporal filter aligns with the actual time more closely. Overall, the results demonstrate that the temporal model maintains meaningful state representations, allowing for capping prediction without direct observations.

5.7. Application

The above experiments demonstrate the reliability of spatial mapping combined with the temporal filter for classifying and predicting cell states in individual cell sequences. The resulting



a) Side 0 b) Side 1

Figure 14: The final comb map, as of September 9th, 2024, visualised over the latest scan for both sides of the honeybee comb. Each cell is highlighted in a color representing the class with the highest belief—*other* (red), *egg* (blue), *larva* (purple), and *capped brood* (green).

map is shown in Fig. 14. However, the primary value of this map lies in its broader application as a tool for studying honeybee colonies and their overall well-being. In this section, we present two example applications—real-time quantification of brood cannibalism and estimation of the colony state through comb monitoring. The system enables us to estimate the occurrence of egg cannibalism—a phenomenon that reflects colony fitness and the amount of food storage—thanks to the detection of egg-laying events and individual cell monitoring over time (Sec. 5.7.1). We can estimate the state of the colony using the counts of individual cell types leveraging the semantic map as a whole (Sec. 5.7.2).

That is, of course, not an exhaustive list. Beyond individual cell tracking, the spatial map also provides insights into the structural dynamics of the comb. For example, analyzing the compactness of the broodnest can help to assess the queen's egg-laying pattern and overall productivity. Irregular or scattered brood patches may indicate queen failure or inadequate mating, both of which are critical factors in long-term colony stability Tarpy et al. (2013); Walsh and Rangel (2016). By detecting deviations in brood patterns early, beekeepers can respond promptly—either by replacing a failing queen or adjusting management practices—reducing the likelihood of cascading problems that might otherwise threaten colony survival. Additionally, it could help analyze the formation of broodnest patterns, which have been previously reported to occur in honeybee colonies Camazine (1991), facilitating early detection of brood diseases Genersch (2010) or parasitic infestations Dietemann et al. (2013), revealing how colonies allocate resources over time and offering insights into self-organizing processes that shape the broodnest Camazine et al. (1990); Johnson (2008); Montovan et al. (2013). Studying the structural dynamics is beyond the scope of this work. We invite our readers to come up with other exciting applications of our work.

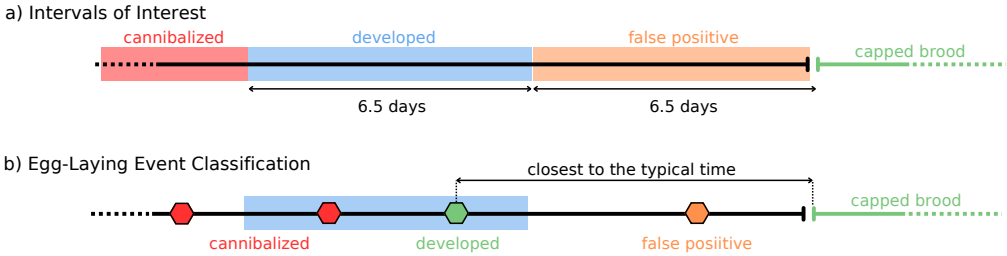


Figure 15: Visualisation of a) intervals of interest for egg-laying classification in egg cannibalism estimation and b) classification of egg-laying events. When a single egg-laying event occurs, it is classified based on the intervals of interest. In cases of multiple egg-laying events, the event with the closest capping time to the typical value is identified, while preceding egg-layings are classified as cannibalized and subsequent ones as false positives.

5.7.1. Estimating Egg Cannibalism

Brood cannibalism in honeybee colonies is a response to environmental stressors, particularly food shortages Schmickl and Crailsheim (2001, 2004). When resources are limited, worker bees consume younger brood to reallocate nutrients toward older larvae, in which they have already invested more resources Brodschneider and Crailsheim (2010). By assessing the egg cannibalism rate, we could identify resource shortages or colony stress that could lead to the potential collapse of the colony.

Since our system detects egg-laying events, we can estimate the rate of egg cannibalism using the final comb map. For each confirmed egg-laying event, we search for the nearest capped brood sequence in the future. If the egg-laying event is not interrupted by another egg-laying before capping and the time interval between the event and capping is greater than 6.5 days and less than 13 days (accounting for potential time error in the temporal model), we consider the egg to have successfully developed. If the time interval between the egg-laying event and the end of the dataset is less than 13 days, egg-laying is not considered in the metrics. Figure 15 gives a diagram of the process with intervals of interest and a schematic visualisation of the classification of egg-laying events.

In cases where multiple egg-laying events precede a single capped brood sequence, we first determine the sex of the brood based on the duration of the capped state. Then, we analyze the time intervals between the egg-laying events and capping to identify the egg-laying event most likely responsible for the capped brood. This egg-laying event is counted as successfully developed. Any egg-laying events that occurred after the identified successful one are omitted, we consider them to be either false positive detections or cases where the queen did not actually deposit an egg in the cell. Conversely, egg-laying events that occurred before the successful one are classified as cannibalized, as they did not lead to brood development despite being detected.

The overall brood cannibalism rate computed by our method over the span of our dataset was 31.4%. Out of 2136 detected egg-laying events, 1466 eggs successfully developed into a capped brood, with a mean and standard deviation of the time between egg-laying and capping being (8.3 ± 1.0) days. This time includes the inherent delay of both the temporal filter and irregular observations. In 75 cases, the detected egg-laying event was either a false positive or a case where the queen did not actually deposit an egg in the cell. Among the 670 cannibalized eggs, 427 were not followed by any capping event, while 207 were cannibalized but later replaced by a new egg-laying event that successfully led to a capped brood. Additionally, 36 cannibalized eggs were

capped later than expected, with a mean time and standard deviation of (14.6 ± 1.3) days, with no other egg-laying detected between, indicating that the actual egg-laying event responsible for the capping may not have been detected due to tracking failures or missed observations.

This relatively high cannibalism rate could be influenced by seasonal factors, as the dataset was collected toward the end of the season when honeybees begin to prepare for winter. Additionally, some detections may correspond to instances in which the queen exhibited egg-laying behaviour without actually depositing an egg in the cell, contributing to the estimated rate. Notably, the honeybee colony remained alive at the time of submitting this article—March 2025, approximately six months after data collection—suggesting that the colony was healthy but rather under resource stress.

5.7.2. Honeycomb State Development

Monitoring the development of a honeybee colony is essential to understand its overall health, reproductive success, and long-term well-being. Potentially, it could even serve as an early warning sign of a threatening colony loss. In this section, we demonstrate the tracking of the brood stages in the hive over time.

Earlier, we have demonstrated the classification performance of the temporal filter (see Section 5.6). By integrating observations over time, the temporal filter enables us to track the development of the entire comb. For each individual cell $c_{t,j}$ at any time t, the filter provides a belief distribution $p(H_{t,j} \mid z_{1:t-1})$ over possible cell states. To show the results at a reasonable granularity, we estimate totals per each defined class, disregarding multiple days per class.

We estimate the total number of cells after integrating observations from each scan of the comb. For each cell $c_{t,j}$, we first sum the probabilities in a way to obtain probabilities $P(H_{t,j} = k \mid z_{1:t-1})$ where class k is one of *other*, *egg*, *larva* or *capped brood*. Now, to get an estimate of the total number of cells per class k, we use the fact that the cells are estimated independently. We consider for each cell $c_{t,j}$ a binary random variable $K_{t,j} \sim Bernouli(p_{t,j})$, where $K_{t,j} = 1$ iff the cell belongs to class k, then trivially $p_{t,j} = P(H_{t,j} = k \mid z_{1:t-1})$. The total expected number of cells for each class at time t is then

$$\mathrm{E}[\sum_i K_{t,i}] = \sum_{i=1}^{N_t} \mathrm{E}[K_{t,i}] = \sum_{i=1}^{N_t} p_{t,i}. \tag{7}$$

and the associated standard deviation to account for the uncertainty of the estimate is similarly

$$\sqrt{\mathrm{Var}[\sum_i K_{t,i}]} = \sqrt{\sum_{i=1}^{N_t} \mathrm{Var}[K_{t,i}]} = \sqrt{\sum_{i=1}^{N_t} p(1-p)}. \tag{8}$$

For cells that have not yet been observed, we consider their distribution to be uniform over the aggregated classes k to make the estimates comparable over time. It should be noted that for this normalization, we use the total number of cells in the last timestamp of the dataset, so the calculation cannot be done in real-time. However, for the long-term deployment of such a mapping system, it is a reasonable assumption that we would let the system build the map until we get sufficient coverage of the entire comb.

Figure 16 illustrates the development of the comb state on both sides of the comb. The period for sides 0 and 1 differs due to the varying scanning frequencies; consequently, the initial map takes longer to build. The figure shows both the estimated totals and their respective variance;

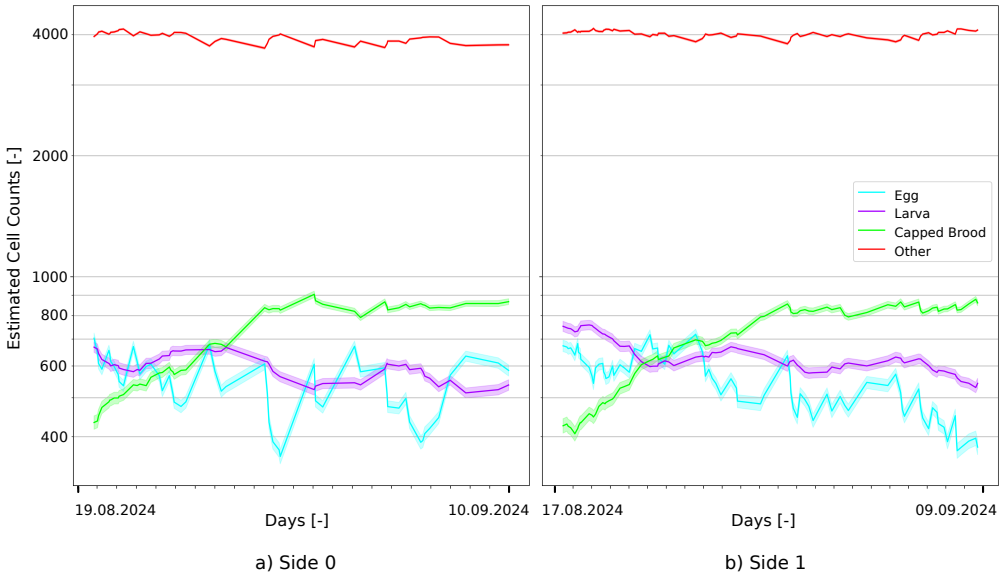


Figure 16: Example application of our system to long-term continuous monitoring of the comb. Starting after the initial map was created, we show the development of the estimated total number of cells for each content class. The estimates are done directly from cell state beliefs with no further filtering. Because only cells that were seen open are included in the map, it takes time before the number of *capped brood* cells is properly stabilised. Note that the y-axis is in a logarithmic scale (base 10).

however, since the total number of cells is in the order of thousands, the variance of expectation is very small (in order of tens) so it is hard to see. The number of *egg* cells exhibits more fluctuations compared to *larva* and *capped brood* cells, which could be attributed to the high cannibalism rate and lower classification accuracy for *egg* cells. In the first few days, we observe a gradual increase in the number of *capped brood* cells, which then stabilizes. This is caused by the fact that we can only track capped brood cells when they were previously detected in their open stage. Cells we did not see are assumed to be uniform, so they contribute to counts of all classes. Nevertheless, the transition from *larva* cells to *capped brood* cells is clearly visible, reflecting the expected developmental progression.

6. Limitations

Despite the promising results of our comb semantic mapping method, several limitations of our study must be acknowledged. Our experimental setup was limited to a single observation hive containing only two combs, which was located indoors with a connection to an outdoor environment via a plastic tube. Although such settings are common in biological research and sufficient for validation, they naturally limit the colony size and may influence its behavioural dynamics. The 25-day duration of our experiment also limited the ability to assess the performance of the brood development model over multiple subsequent brood cycles.

The data collection was non-invasive, meaning that the monitoring was performed under near-infrared light and without any manipulation of the comb frames. However, on 21 August 2024,

the honeybee queen had to be re-tagged, and the colony received a treatment against Varroa mite. We attempted to minimize any disturbance to the colony, but it should be noted that such operations could have introduced minor artifacts or deviations in colony behavior.

The AROBA system was specifically developed for long-term observation and interaction with a honeybee queen in a laboratory-controlled environment, primarily for biological research. As such, the used mechatronic solution—due to its size, complexity, and reliance on a non-standard observation hive with a limited-size colony—is not suitable for practical deployment in standard hives and would be economically impractical for traditional beekeeping at scale.

Nevertheless, we believe that our mapping system provides valuable data for biological and ecological research, where long-term data of the comb dynamics is usually difficult to collect. We envision our system in its current state as a research instrument rather than a field-deployable solution.

7. Future Work

In the future, we plan to design a more robust way of comb scan registration that would not require precise odometry information from the robot. As we observe the colony in its natural environment under near-infrared light, it may also be beneficial to explore computer vision methods for enhancing image clarity through colorization, such as Shalini et al. (2024), as well as considering alternative object detection approaches Vasantha et al. (2023). A big hurdle is also setting the parameters for the temporal model, which further may be dependent on the specific colony, weather conditions, time of the season, and others; we would like to be able to learn the parameters from the data . An important property of honeybee hives is their spatial organisation. As we assume that the cells are developing independently, modelling spatial dependencies directly could be another extension of our method . One of the important factors influencing the state of the comb is the behavior of the honeybee queen. In Blaha et al. (2025) , we show that the queen movement does not correspond to any simple random walk models; therefore, beyond modelling spatial dependencies between individual cells, future work could also incorporate other queen behavioral data, such as her trajectory or resting locations, in the semantic map. While improving the technical level of our method, we hope to reach more novel applications. We see great potential in the study of the honeybee behavior, both under natural conditions and under stressful interventions, such as nutritional deprivation, presence of agricultural chemicals or brood diseases.

8. Conclusion

Monitoring the health and state of honeybee colonies is crucial for detecting early signs of colony stress. One option for such monitoring is continual observation of the comb, which contains information about the colony development, food storage, etc. Most existing methods, unfortunately, require manual intervention by experienced beekeepers, which is not scalable to allow for more than one-time observations and also disrupts the colony. In this work, we presented a proof-of-concept for a non-invasive continuous mapping of comb brood cells using robotic observations from within an indoor-located observation hive.

We successfully managed to build and automatically update a semantic map of the comb despite heavy occlusions caused by honeybees and demonstrated two potential applications of our work. The first is a continuous colony health assessment by measuring the brood cannibalism

rate. To our knowledge, it is the first system with such automatic capability. And the second is the monitoring of the comb development and overall state estimation in real time. While the current robotic platform is not yet applicable to practical beekeeping, we aim to design a system deployable with standard hives in the future.

8.1. Contributions

8.2. Future Work

Appendix A.

Appendix A.1. Full Transition Model Diagram

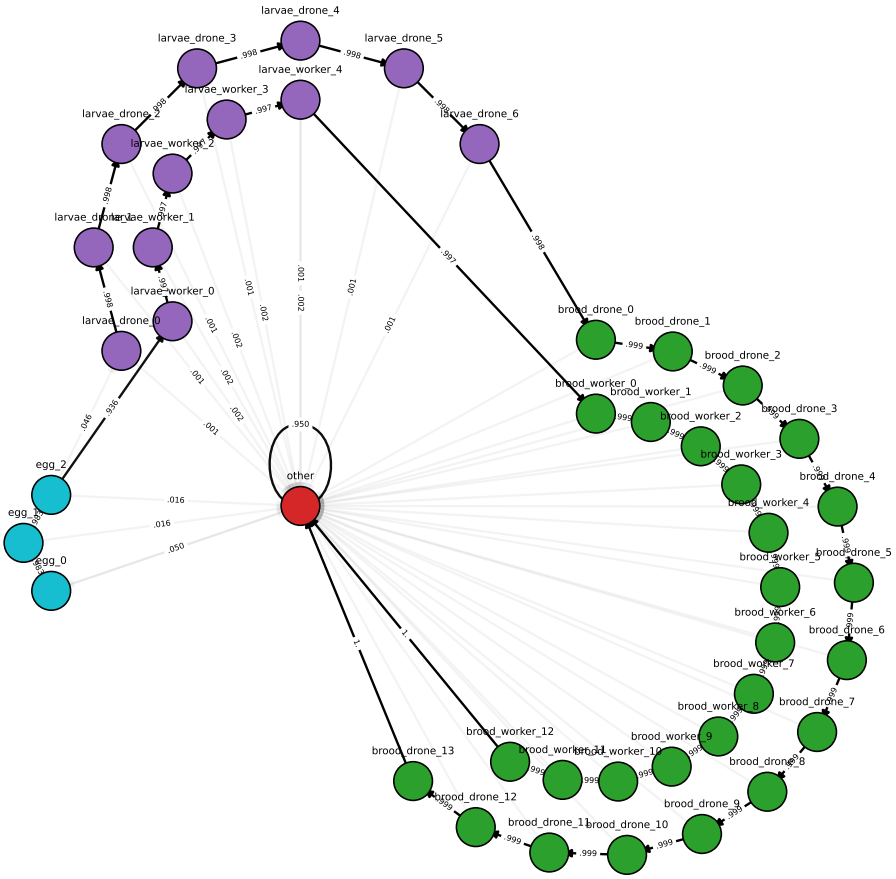


Figure A.17: A graph visualization of the considered cell states and the transition model describing the development of brood. Individual nodes correspond to the possible states of the cells—*egg* (blue), worker/drone *larva* (orange), worker/drone *capped brood* (red) and *other* (green). To achieve the Markovian property, the differences in length of individual stages—also in connection to the sex of the brood—are represented by internal hidden states per each day (based on literature review).

Appendix A.2. System Odometry Quality

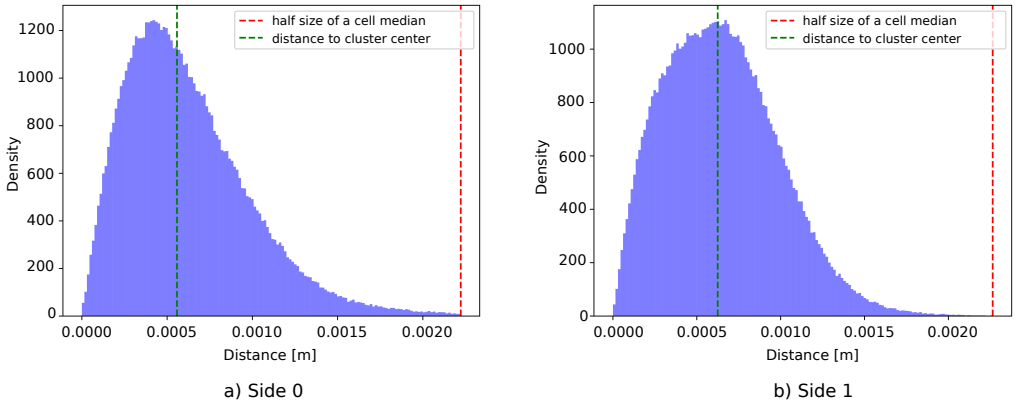


Figure A.18: Distributions of detection-to-cluster distances for both sides of the comb, which we use to check the odometry error for each robot. The mass of both distributions is concentrated well below half the size of a cell, which means that detections of the same cell should be reliably matched together.

Appendix A.3. Cell Image Dataset

Class	Training	Validation	Test
Other	13 954	3534	8963
Egg	263	90	192
Larva	757	291	599
Capped brood	7488	3166	6368
Any open brood	304	123	281
Occluded/capped honey	62 744	17 652	39 067

Table A.10: Number of annotated cell images in each class for training, validation, and test sets. Classes corresponding to states are shown in italics.

Acknowledgements

This work was supported by the EU HORIZON-EIC-2023-PATHFINDER-OPEN-01 project SensorBees number 101130325, EU H2020-FET-OPEN RoboRoyale project number 964492, project Robotics and Advanced Industrial Production no. CZ.02.01.01/00/22_008/0004590, and by the Grant Agency of the Czech Technical University project SGS22/168/OHK3/3T/13, TS, LAF, MS were supported by the Field of Excellence COLIBRI (Complexity of Life in Basic Research and Innovation) of the University of Graz.

Data and Code Availability

A part of the data collected and analyzed during the current study is available at `https://owncloud.roboroyale.eu/s/IJBhlu5rjKpmOFB` for the purpose of manuscript review.

This data will be published in a public repository for final submission. Additional datasets are available upon a reasonable request. The code for the analysis performed in our study is provided at `https://gitlab.roboroyale.eu/janota/2025_cea_mapping`.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the authors used ChatGPT, Writefull and Grammarly in order to improve the language and readability of parts of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Experimental Conditions and Hive Management

The (*Apis mellifera* L.) colony was originally bred by Dr. Norbert Hrassnigg (Viktring 9073, Austria) and has been maintained by professional beekeepers under standardized conditions at the University of Graz since March 2024. During the observational period, the colony was maintained as it would be under normal beekeeping conditions. Standard apicultural practices were applied throughout the study, which included periodic prophylactic or therapeutic measures against potential Varroa infestations. These treatments were part of routine hive management, carried out independently of any experimental observations.

No special criteria were set to include or exclude any animals from this observational study. To minimize disturbance, visual inspections were performed through the hive's transparent glass, reducing the need for direct handling; any essential manipulations (such as regular glass exchange) were carried out using standard beekeeping procedures. The queen bee was marked with a shellac-based adhesive following established apicultural practices.

No euthanasia or additional interventions were required at the end of the observational period. The colony remains active and continues to thrive in the same hive. Although insects are exempt from formal ethical approvals under the Austrian Animal Experiments Act (TVG 2012, 1. Abs., §1), the methods and reporting in this study were guided by relevant principles of the ARRIVE (Animal Research: Reporting of In Vivo Experiments) guidelines to ensure clarity and reproducibility.

CRediT Authorship Contribution Statement

Conceptualization: JJ, JB, TK. Methodology: JJ, JB, TK. Software: JJ, JB, TK. Validation: JJ. Formal analysis: JJ, JB. Investigation: JJ, JU, TR, LF, MS, FRB. Resources: TK, FA, TS. Data curation: JJ, JU, TR, LF, TK. Writing - original draft: JJ, JB, TK, MS. Writing - review & editing: JJ, JB, JU, TR, LF, TK, MS, FRB, FA, TS. Visualization: JJ, JB. Supervision: TK. Project administration: TK, FA, TS. Funding acquisition: TK, FA, TS.

References

Allsopp, M.H., De Lange, W.J., Veldtman, R., 2008. Valuing insect pollination services with cost of replacement. PLoS ONE 3, e3128. doi:10.1371/journal.pone.0003128.

Alves, T.S., Pinto, M.A., Ventura, P., Neves, C.J., Biron, D.G., Junior, A.C., De Paula Filho, P.L., Rodrigues, P.J., 2020. Automatic detection and classification of honey bee comb cells using deep learning. Computers and Electronics in Agriculture 170, 105244. doi:10.1016/j.compag.2020.105244.

Barmak, R., Stefanec, M., Hofstadler, D.N., Piotet, L., Schönwetter-Fuchs-Schistek, S., Mondada, F., Schmickl, T., Mills, R., 2023. A robotic honeycomb for interaction with a honeybee colony. Science Robotics 8, eadd7385. doi:10.1126/scirobotics.add7385.

Beewise Technologies Ltd., 2025. Beewise – saving bees to feed the world. https://beewise.ag. Accessed: 2025-06-03.

Blaha, J., Stefanec, M., Janota, J., Hofstadler, D.N., Rouček, T., Ulrich, J., Fedotoff, L.A., Broughton, G., Vintr, T., Arvin, F., Schmickl, T., Krajník, T., 2025. On the movement of the honeybee queen in the hive. Scientific Reports 15, 20708. doi:10.1038/s41598-025-07093-4.

Blaha, J., Vintr, T., Mikula, J., Janota, J., Rouček, T., Ulrich, J., Rekabi-Bana, F., Fedotoff, L.A., Stefanec, M., Schmickl, T., Arvin, F., Kulich, M., Krajník, T., 2024. Toward perpetual occlusion-aware observation of comb states in living honeybee colonies, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 5948–5955. doi:10.1109/IROS58592.2024.10801380.

Borlinghaus, P., Gülzow, J.M., Odemer, R., 2024. In-hive flatbed scanners for non-destructive, long-term monitoring of honey bee brood, pathogens and pests. Smart Agricultural Technology 9, 100655. doi:10.1016/j.atech.2024.100655.

Bozek, K., Hebert, L., Portugal, Y., Mikheyev, A.S., Stephens, G.J., 2021. Markerless tracking of an entire honey bee colony. Nature Communications 12, 1733. doi:10.1038/s41467-021-21769-1.

Brodschneider, R., Crailsheim, K., 2010. Nutrition and health in honey bees. Apidologie 41, 278–294. doi:10.1051/apido/2010012.

Camazine, S., 1991. Self-organizing pattern formation on the combs of honey bee colonies. Behavioral Ecology and Sociobiology 28, 61–76. doi:10.1007/BF00172140.

Camazine, S., Sneyd, J., Jenkins, M.J., Murray, J., 1990. A mathematical model of self-organized pattern formation on the combs of honeybee colonies. Journal of Theoretical Biology 147, 553–571. doi:10.1016/S0022-5193(05)80264-4.

Colin, T., Bruce, J., Meikle, W.G., Barron, A.B., 2018. The development of honey bee colonies assessed using a new semi-automated brood counting method: CombCount. PLoS ONE 13, e0205816. doi:10.1371/journal.pone.0205816.

Dainat, B., Dietemann, V., Imdorf, A., Charrière, J.D., 2020. A scientific note on the 'Liebefeld Method' to estimate honey bee colony strength: Its history, use, and translation. Apidologie 51, 422–427. doi:10.1007/s13592-019-00728-2.

Defèr, A., 2022. Classification of Honeybee Larval Stages Using CNNs Applied to Image Data. Master's thesis. Freie Universität Berlin.

Dietemann, V., Nazzi, F., Martin, S.J., Anderson, D.L., Locke, B., Delaplane, K.S., Wauquiez, Q., Tannahill, C., Frey, E., Ziegelmann, B., Rosenkranz, P., Ellis, J.D., 2013. Standard methods for varroa research. Journal of Apicultural Research 52, 1–54. doi:10.3896/IBRA.1.52.1.09.

Emsen, B., 2006. Semi-automated measuring capped brood areas of honey bee colonies. Journal of Animal and Veterinary Advances 5, 1229–1232.

Genersch, E., 2010. Honey bee pathology: Current threats to honey bees and beekeeping. Applied Microbiology and Biotechnology 87, 87–97. doi:10.1007/s00253-010-2573-8.

Groeneveld, J., Odemer, R., Requier, F., 2024. Brood indicators are an early warning signal of honey bee colony loss—a simulation-based study. PLoS ONE 19, e0302907. doi:10.1371/journal.pone.0302907.

Gullan, P.J., Cranston, P.S., 2014. Insects: An Outline of Entomology. 5th ed., Wiley Blackwell, Chichester.

Hadjur, H., Ammar, D., Lefèvre, L., 2022. Toward an intelligent and efficient beehive: A survey of precision beekeeping systems and services. Computers and Electronics in Agriculture 192, 106604. doi:10.1016/j.compag.2021.106604.

Höferlin, B., Höferlin, M., Kleinhenz, M., Bargen, H., 2013. Automatic analysis of *Apis mellifera* comb photos and brood development, in: Association of Institutes for Bee Research Report of the 60th Seminar in Würzburg. Apidologie 44, p. 19.

Illingworth, J., Kittler, J., 1987. The adaptive hough transform. IEEE Transactions on Pattern Analysis and Machine Intelligence , 690–698doi:10.1109/TPAMI.1987.4767964.

Janota, J., Blaha, J., Rekabi-Bana, F., Ulrich, J., Stefanec, M., Fedotoff, L., Arvin, F., Schmickl, T., Krajník, T., 2024. Towards robotic mapping of a honeybee comb, in: 2024 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS), pp. 1–6. doi:10.1109/MARSS61851.2024.10612712.

Jeffree, E.P., 1951. A photographic presentation of estimated numbers of honeybees (*Apis mellifera* L.) on combs in 14 × 8½ inch frames. Bee World 32, 89–91. doi:10.1080/0005772X.1951.11094703.

Jeffree, E.P., 1958. A shaped wire grid for estimating quantities of brood and pollen in combs. Bee World 39, 115–118. doi:10.1080/0005772x.1958.11095048.

Jocher, G., et al., 2022. Ultralytics/yolov5: v7.0 - YOLOv5 SOTA realtime instance segmentation. Zenodo. doi:10.5281/zenodo.3908559.

Johnson, B.R., 2008. Pattern formation on the combs of honeybees: Increasing fitness by coupling self-organization with templates. Proceedings of the Royal Society B: Biological Sciences 276, 255–261. doi:10.1098/rspb.2008.0793.

Kevan, P.G., Viana, B.F., 2003. The global decline of pollination services. Biodiversity 4, 3–8. doi:10.1080/14888386.2003.9712703.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al., 2023. Segment anything, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4015–4026.

Knauer, U., Himmelsbach, M., Winkler, F., Zautke, F., Bienefeld, K., Meffert, B., 2005. Application of an adaptive background model for monitoring honeybeesp, in: Visualization, Imaging, and Image Processing (VIIP).

Liew, L.H., Lee, B.Y., Chan, M., 2010. Cell detection for bee comb images using Circular Hough Transformation, in: 2010 International Conference on Science and Social Research (CSSR), pp. 191–195. doi:10.1109/CSSR.2010.5773764.

Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context, in: Computer Vision – ECCV 2014, pp. 740–755. doi:10.1007/978-3-319-10602-1_48.

Meikle, W.G., Holst, N., 2014. Application of continuous monitoring of honeybee colonies. Apidologie 46, 10–22. doi:10.1007/s13592-014-0298-x.

Milovanović, M., Pejić, J., Pejić, P., 2025. Advanced sensors for noninvasive bee colony inspection. Computers and Electronics in Agriculture 231, 109945. doi:10.1016/j.compag.2025.109945.

Montovan, K.J., Karst, N., Jones, L.E., Seeley, T.D., 2013. Local behavioral rules sustain the cell allocation pattern in the combs of honey bee colonies (*Apis mellifera*). Journal of Theoretical Biology 336, 75–86. doi:10.1016/j.jtbi.2013.07.010.

Paolillo, G., Petrini, A., Casiraghi, E., De Iorio, M.G., Biffani, S., Pagnacco, G., Minozzi, G., Valentini, G., 2022. Automated image analysis to assess hygienic behaviour of honeybees. PLoS ONE 17, e0263183. doi:10.1371/journal.pone.0263183.

Paudel, Y.P., Mackereth, R., Hanley, R., Qin, W., 2015. Honey bees (*Apis mellifera* L.) and pollination issues: Current status, impacts, and potential drivers of decline. Journal of Agricultural Science 7, 93. doi:10.5539/jas.v7n6p93.

Potts, S.G., Biesmeijer, J.C., Kremen, C., Neumann, P., Schweiger, O., Kunin, W.E., 2010. Global pollinator declines: Trends, impacts and drivers. Trends in Ecology & Evolution 25, 345–353. doi:10.1016/j.tree.2010.01.007.

Rathore, N., Tyagi, P.K., Agrawal, D., 2023. Semi-automatic analysis of cells in honeybee comb images, in: 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), pp. 1–4. doi:10.1109/SCEECS57921.2023.10063122.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 1137–1149. doi:10.1109/tpami.2016.2577031.

Rodrigues, P.J., Neves, C.J., Pinto, M.A., 2016. Geometric contrast feature for automatic visual counting of honey bee brood capped cells, in: EURBEE 2016: 7th European Conference of Apidology.

Rodriguez-Lozano, F.J., Geninatti, S.R., Flores, J.M., Quiles-Latorre, F.J., Ortiz-Lopez, M., 2024. Capped honey segmentation in honey combs based on deep learning approach. Computers and Electronics in Agriculture 227, 109573. doi:10.1016/j.compag.2024.109573.

Romano, D., Porfiri, M., Zahadat, P., Schmickl, T., 2024. Animal–robot interaction—an emerging field at the intersection of biology and robotics. Bioinspiration & Biomimetics 19, 020201. doi:10.1088/1748-3190/ad2086.

Schmickl, T., Crailsheim, K., 2001. Cannibalism and early capping: Strategy of honeybee colonies in times of experimental pollen shortages. Journal of Comparative Physiology A 187, 541–547. doi:10.1007/s003590100226.

Schmickl, T., Crailsheim, K., 2004. Inner nest homeostasis in a changing environment with special emphasis on honey bee brood nursing and pollen supply. Apidologie 35, 249–263. doi:10.1051/apido:2004019.

Shaheen, F.A., Khan, K.A., Husain, M., Mahmood, R., Rafique, M.K., 2017. Role of honey bees (*Apis mellifera* L.) foraging activities in increased fruit setting and production of apples (*Malus domestica*). Pakistan Journal of Agricultural Research 30.

Shalini, G.M., Anand, R., Sankeerth, T., Reddy, H.V., Hariharan, S., Kukreja, V., 2024. Integrated colorization and styling approach for enhancing image clarity, in: 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp. 798–804. doi:10.1109/ICAAIC60222.2024.10575041.

Sparavigna, A.C., 2016. Analysis of a natural honeycomb by means of an image segmentation. Philica 2016, hal–01416832.

Stefanec, M., Hofstadler, D.N., Kärcher, M., Schmickl, T., 2025. Broodscan: A honeybee brood scanner technology. bioRxiv doi:10.1101/2025.05.17.651227.

Tarpy, D.R., Lengerich, E.J., Pettis, J.S., 2013. Idiopathic brood disease syndrome and queen events as precursors of colony mortality in migratory beekeeping operations in the eastern United States. Preventive veterinary medicine 108, 225–233. doi:10.1016/j.prevetmed.2012.08.004.

Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic Robotics. The MIT Press, Cambridge, Massachusetts, USA.

Ulrich, J., Blaha, J., Alsayed, A., Rouček, T., Arvin, F., Krajník, T., 2023. Real time fiducial marker localisation system with full 6 dof pose estimation. ACM SIGAPP Applied Computing Review 23, 20–35. doi:10.1145/3594264.3594266.

Ulrich, J., Stefanec, M., Rekabi-Bana, F., Fedotoff, L.A., Rouček, T., Gündeğer, B.Y., Saadat, M., Blaha, J., Janota, J., Hofstadler, D.N., et al., 2024. Autonomous tracking of honey bee behaviors over long-term periods with cooperating robots. Science Robotics 9, eadn6848. doi:10.1126/scirobotics.adn6848.

Vasantha, S., Kiranmai, B., Hussain, M.A., Hashmi, S.S., Nelson, L., Hariharan, S., 2023. Face and object detection algorithms for people counting applications, in: 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), pp. 1188–1193. doi:10.1109/ICACRS58579.2023.10405114.

Walsh, E., Rangel, J., 2016. Local honey bee queen production and quality. Bee World 93, 30–32. doi:10.1080/0005772X.2016.1211501.

Yoshiyama, M., Kimura, K., Saitoh, K., Iwata, H., 2011. Measuring colony development in honey bees by simple digital image analysis. Journal of Apicultural Research 50, 170–172. doi:10.3896/IBRA.1.50.2.10.