CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

BACHELOR THESIS

Kristi Žampachů

Visual analysis of beehive queen behaviour

Department of Computer Science

Thesis supervisor: **Ing. Tomáš Krajník, PhD**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Žampachů** Jméno: **Kristi** Osobní číslo: **495641**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**

Studijní program: **Otevřená informatika**

Specializace: **Počítačové hry a grafika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Visual analysis of beehive queen behaviour

Název bakalářské práce anglicky:

Systém pro analýzu chování včelí královny

Pokyny pro vypracování:

The aim of the project is to implement a system for automatic detection and localisation of the honeybee queen and analysis of her movement. The system should enable basic analysis of the honeybee queen's interactions with her surroundings.
1) Learn (from the literature) about the basics of the honeybee behaviours and the interactions that within the beehive colony.
2) Learn about the systems capable of automated detection, tracking and behaviour analysis of the honeybees within the beehive.
3) Establish a set of key performance indicators (KPI) that characterize the performance of the aforementioned systems in the context of the project aim.
4) Select the most relevant method(s) and extend them so their performance in queen and court detection can be assessed.
5) Assess the method(s) performance using the selected KPIs and discuss the results.
6) Based on the results, integrate your method into the pipeline for honeybee queen detection and court behaviour analysis.

Seznam doporučené literatury:

[1] Bozek, Katarzyna, Laetitia Hebert, Yoann Portugal, Alexander S. Mikheyev, and Greg J. Stephens. 'Markerless tracking of an entire honey bee colony.' Nature communications 12, no. 1 (2021): 1-13.
[2] Schmickl et al.: The Queen and her Robotic Court: A Minimally-Invasive Form of Ecosystem Hacking: In International Science Fiction Prototyping Conference (SciFi-It' 2021).
[3] Boenisch, Franziska, Benjamin Rosemann, Benjamin Wild, David Dormagen, Fernando Wario, and Tim Landgraf. 'Tracking all members of a honey bee colony over their lifetime using learned models of correspondence.' Frontiers in Robotics and AI 5 (2018): 35.
[4] Redmon, Joseph, and Ali Farhadi. 'Yolov3: An incremental improvement.' arXiv preprint arXiv:1804.02767 (2018).

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Tomáš Krajník, Ph.D. centrum umělé inteligence FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **04.02.2022** Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

doc. Ing. Tomáš Krajník, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

.

Datum převzetí zadání

Podpis studenta

Declaration

I hereby declare that I have completed this thesis independently and that I have used only the sources (literature, software, etc.) listed in the enclosed bibliography.

In Prague on.............................   ....

Acknowledgements

I would like to thank my family for the continuous support during my studies. Also I thank to my supervisor Ing. Tomáš Krajník, PhD, for the opportunity to work on this topic and for being a great mentor. This work was supported by the EU FET Open programme under grant agreement No.964492 project 'RoboRoyale'.

Abstrakt

Tato práce se zabývá detekcí včelí královny uvnitř včelího úlu. Z metod na detekci a sledování objektů jsme vybrali WhyCode, který nejlépe splňuje naše požadavky, výpočetně nenáročnou detekci a přesný odhad pozice. V práci navrhujeme nový systém, WhyComb, který vychází z WhyCode. WhyComb implementuje filtry na odchytávání falešně pozitivních detekcí. Dále přidává detekci pomocí konvoluce, která detekuje královnu při okluzi WhyCode markeru. Oba systémy jsme porovnávali pomocí 'precision', 'recall' a Wilcoxonova párového testu. Výsledky ukazují že WhyComb detekuje královnu častěji s menší průměrnou chybou detekce. Whycomb je součástí projektu 'RoboRoyale', který má za cíl integrování robotických včel do dvoru královny.

Abstract

This thesis deals with detecting the honey bee queen inside the hive. From different detection and tracking methods, we selected Whycode, which best met our requirements, computationally efficient detection and precise position estimation. Here we propose a new system, WhyComb, that is built upon WhyCode. Whycomb implements filters to filter out false-positive detections. Futhermore, it adds convolution-based detection, which is used in where the WhyCode marker is occluded. We compared both systems with precision, recall, and Wilcoxon pairwise test. The results show that WhyComb detects the queen more often and has a lower average detection error. WhyComb is part of a larger project, 'RoboRoyale', that aims to integrate robotic bees into the queen's court.

Contents

List of Figures

List of Tables

1 Introduction

Honey bees are vital to the earth's ecosystem. They are the primary pollinators of flowering plants in both agricultural and natural ecosystems. Their contribution to the economy is immense, estimated at Fourteen billion dollars in the US [8], by pollinating important crops, including apples, coffee, pumpkins , and tomatoes. The bees are able to pollinate plants in an area of up to 280 km2 around their beehive, with tens of thousands of pollination flights per day [9]. In natural ecosystems, bees advance the reproduction of plants and the spread of vegetation and, in turn, provide food for animals [2].

This all is possible because the honey bees are social insects, the beehive acting as a singular superorganism. At the centre of this organism is the honey bee queen, who is influencing the whole colony. The main form of interaction and communication between honey bees is pheromones. The primary pheromone created by the queen is the queen mandibular pheromone (QMP). It is responsible for inducing 'attraction' to the queen and the creation of the queen's retinue (the queen's court) that is responsible for feeding and grooming the queen [10]. It also attracts drones to her during mating flight and inhibits the development of ovaries in workers, and this is so that no new queen will rise [11].

Despite the queen being the primary coordinator, the worker bees are able to achieve a coordinated activity between tens of thousands of bees[12]. The instance where it is most visible is food gathering with the waggle dance. It is a form of interaction between a successful forager bee that came back to the hive and other forager bees inside the hive. It consists of movement and odour information. The dancer bee dances in an 8-like shape to the sides and a forward waggle in the middle, see Figure 1 and paper [1]. The dances occur at the honeycombs, which means that the dancer can not use their head to point in the direction of the food source. Instead, they use the orientation of the body of the dancer relative to gravity to indicate the direction. The duration of the waggle in the middle of the dance pattern tells other foragers the distance from the hive. The longer the waggle, the further the food source is. The onlooking bees can also use the odour of the food that the dancer brought in [13].

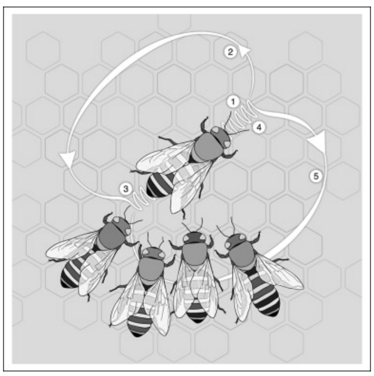


Figure 1: Visualization of a bee performing the waggle dance [1]

As stated above, the main recipients of the communication from the queen are the worker bees in the queen's court. These worker bees are the bridge between the queen and the hive and so have a direct influence on the queen. The RoboRoyale project aims to integrate a set of biomimetic robots into the queen's court to influence the queen and subsequently on the whole beehive [2]. This integration would allow for optimising the hive's macroscopic variables by regulating the queen's pheromone production.

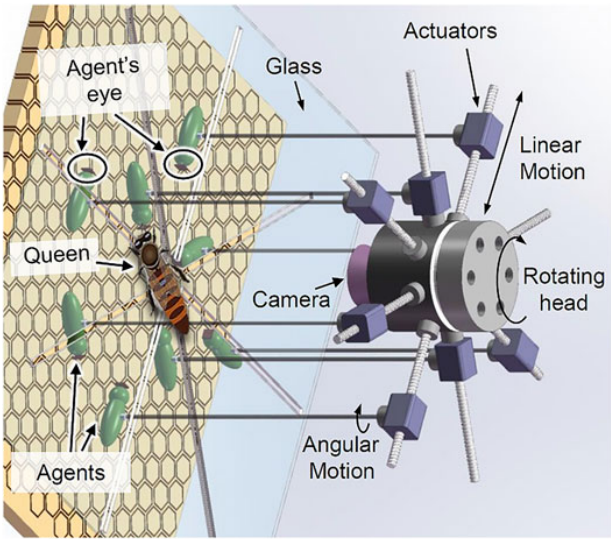


Figure 2: Basic concept from a physical robotic system.[2]

To be able to form the court around the queen reliably, we need to be able to detect and track the queen with the robotic system presented in Figure 2. This thesis aims to give an overview of commonly used detection systems and to extend and specialise one so that it can reliably track the queen's position on the comb of the beehive.

The thesis is divided into four chapters. Firstly, an introduction to state-of-the-art methods used for tracking and detection, with a focus on systems that were deployed on honey bees. Then an overview of extensions of the relevant methods. The penultimate chapter describes data collection and used datasets. In the final chapter are the experiments and evaluations of the used methods.

2 State of the art systems

Honey bees are a popular observation target. Mainly it is to learn about the inner workings of the honey bee colony. From how they communicate to how they respond to climate change. Honey bees are globally crucial for food production, and the observation of the colony helps us to monitor the health of the beehive.

The observation is done in observation hives, which are a simplified version of the classic beehive. Observation hives are usually two honeycombs placed vertically, one on top of the other, enclosed by two glass panels, as can be seen in Figure 3. The honeycombs can be 3D printed to ease the establishment of new observation hives. The observation hive is sealed from the top and bottom, and the bees have a designated way out of the hive for foraging. From each side are recording infrared cameras and infrared lights to illuminate the combs see Figure 4.

Figure 3: One side of the observation hive

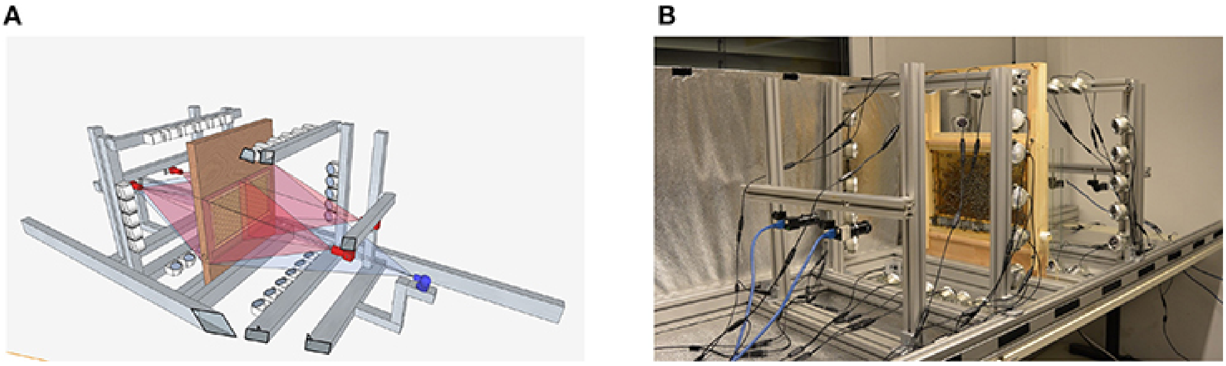


Figure 4: Schematics and realisation of the observation hive[3].

The usage of the infrared lights and cameras is vital because honey bees do not see infrared light, but the cameras do, so this setup allows observation without disturbing the colony. The captured data from the camera is then sent to a computer, and the image is stored, usually as a matrix of values indicating brightness. These matrices are the backbone of computer vision as they are the virtual representation of the real world and are used by a myriad of computer vision methods.

Recognition Recognition is the method for classifying an object in an image from a predefined set of classes.

Localisation Localisation is the method for determining the object's position in the frame coordinates and in real world coordinates.

Detection Detection is a combination of recogntion and localisation used when we do not know the number of objects in the image.

Semantic segmentation Semantic segmentation is the method for classifying all pixels of an image to a class based on its context, multiple objects of the same class are treated as a single entity.

Instance segmentation Instance segmentation is the method for classifying all pixels of an image to a class so each instance of the same class is treated as a separate object. [14]

2.1 Short introduction to vision-based object detection

The goal of vision-based object detection is to locate an object on a given image. This can be achieved by engineered image analysis methods, which process the image in a sequence of steps, often consisting of preprocessing, segmentation, identification and localisation. However, modern machine-learning-based methods can process the image in an end-to-end fashion [15].

Preprocessing is the act of preparing an image for image analysis. It prepares the image to satisfy the input requirements of the detection system. Preprocessing can include noise and outlier removal, filtering background out, changing the image's dimensions or converting the image to greyscale.

Segmentation is the process of dividing the image into multiple image segments, thus simplifying the image. For example in [16] the segmentation is done by thresholding the pixels so the pixel is black or white and then flood-filling neighbouring black pixels, creating segments that can be further processed.

Identification is extracting a set of visual features and then using these features to decide on the class of the object. Visual features can range from the size and shape to the local convexity of the object [17].

Localisation determines the object's position in the image and global coordinate systems. [16] localises the segment inside the image by calculating the centre of the segment from the mean of its pixels' positions. The coordinates in the canonical camera frame are obtained from the positions of canonical vertices and co-vertices. Finally, the segment is transformed into the global coordinates using homography transformation calculated from the positions of the four calibration markers.

The two types of systems introduced in this section are markerless systems (YOLO) and marker-based systems (WhyCode, ArUco).

2.1.1 YOLOv3 algorithm

YOLOv3 (You Only Look Once version three) is a system based on a convolutional neural network. The YOLOv3 network needs to be trained first to be able to detect any specific objects.

The convolutional neural network takes a 416x416 pixel image and returns a tensor with the coordinates of a bounding box, the probability that the bounding box contains one of the trained objects, and the class probabilities of the object inside the bounding box for each class that is being detected. The network is capable of simultaneously detecting multiple objects in one input image. Detections with a probability of detection less than 0.5 are discarded. The detections that are of the same object are filtered out with the nonmax suppression algorithm [4].

2.1.2 WhyCode

WhyCon [16], and its extension WhyCode [5], is a localisation system designed for computationally efficient localisation of a large number of circular black and white fiducial markers of known diameter.

The detection algorithm incorporates flood fill segmentation and on-demand thresholding to pick out possible marker locations, which are then subjected to more rigorous tests

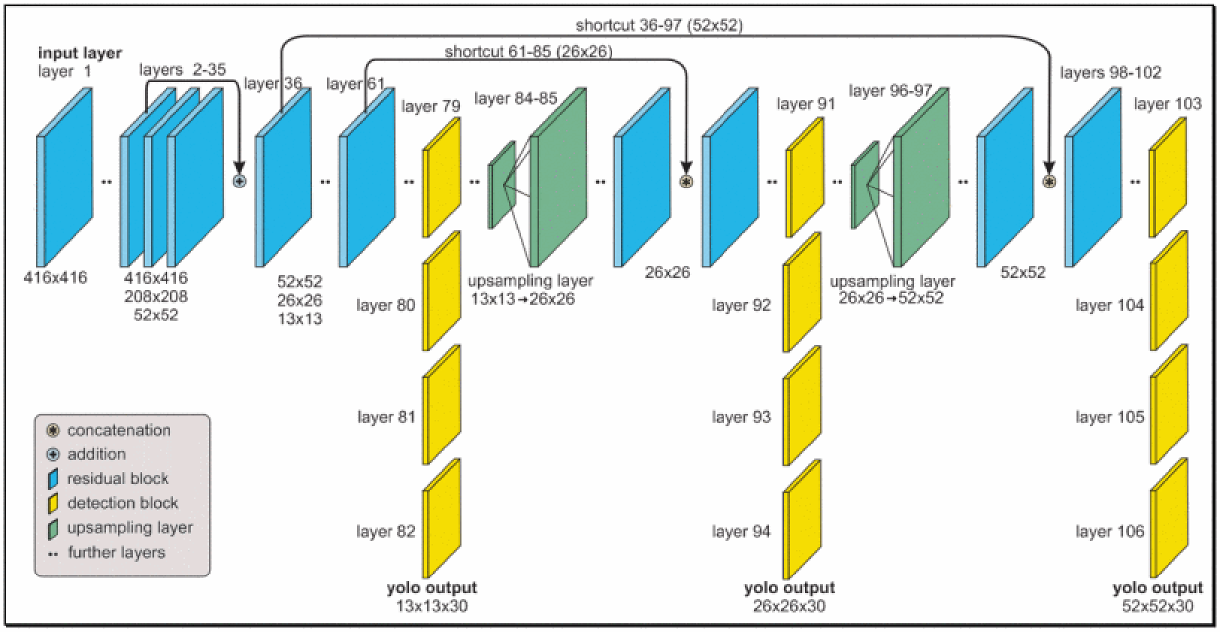


Figure 5: [4, YOLOv3 neural network consist of 106 layers. Besides using convolutional layers, its architecture also contains residual layers, upsampling layers, and skip (shortcut) connections.]

to determine if it is the marker. The algorithm begins with searching for the inner black circle of pixels by finding black pixels and flood filling the segment of black pixels. The algorithm searches for the inner white circle if the segment passes the roundness tests. Concentricity and area ratio tests are performed when the white segment is found. Should these tests be successful, the pattern is considered to be found, and the algorithm will use its position as the starting point for detection on the next frame. A pixel is classified as black or white by the threshold value t, which is adaptively set after each detection. If successful, threshold t is set to the average of the means of the inner and outer circle segment [16].

WhyCode extended WhyCon by identifying the individual markers with an encoding necklace. One of the strengths of the necklace is its invariance to rotations. To get the marker's identification number, the detected encoding has to be bit-shifted to the lowest value. The number of times the encoding was bit-shifted also gives us the rotation of the marker [5]. The necklace is encoded in Manchester encoding to allow for six degrees of freedom position estimation [18, 19].

To get the marker's location, WhyCode transforms the centre of the segment and its eigenvectors into the canonical camera coordinate system. The coefficients of the ellipse's characteristic equation are calculated from the transformed parameters. Eigenvalue analysis is then used on the characteristic equation to obtain the relative position and orientation of the detection, which are then converted into the global coordinate system [16].

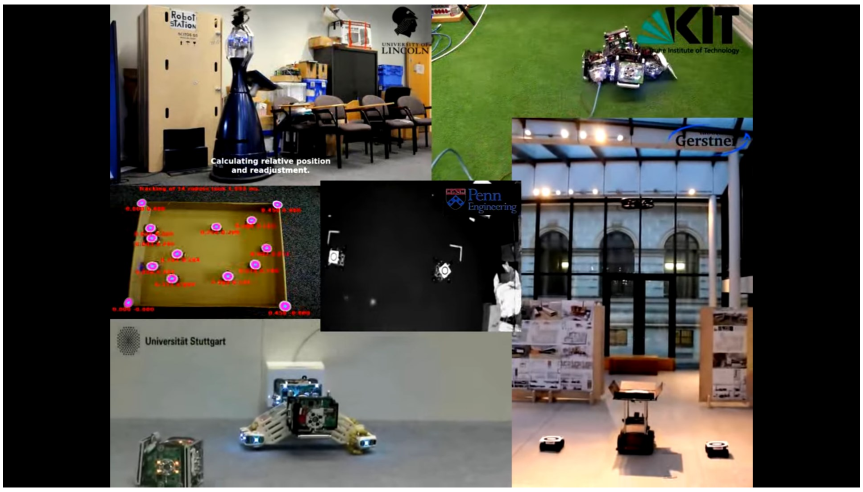


Figure 6: Various situations where the WhyCon marker is used, mainly in robotics.

Tracking in WhyCode is done by repeated re-detection of the marker in real-time. If the marker was detected in the previous frame, then WhyCode will begin its search for the marker at the location of the last detection [16].

2.1.3 ArUco

ArUco is a square-based fiducial marker system. The marker consists of a black square enclosing a binary matrix that encodes its identification code, and it has redundant bits for error correction. In the detection process of the ArUco marker, adaptive thresholding is used to extract contours from a greyscale image. A polygonal approximation is then used on the contours, and then only the ones with an approximation of a 4-vertex polygon are kept [20].

The identification of the ArUco marker is done by analysing the inner region of the detected areas by removing perspective projection and transforming the area into a square. The square is binarized and divided into a regular grid. If all the squares on edge are black, the code inside the area is compared against markers in a dictionary of possible markers [20]. If the code exist in the dictionary then the area is a valid ArUco marker.

To estimate the ArUco marker's relative position by [20, iteratively minimizing the re-projection error of the corners] through the use of the Levenberg-Marquardt algorithm[21].

Similarly to WhyCode, the tracking of the ArUco marker is done by re-detection on each subsequent frame.

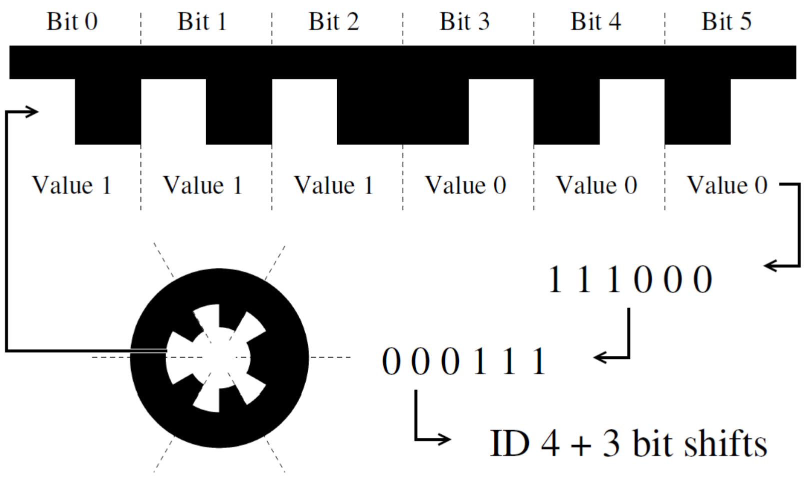


Figure 7: An example of how the Manchester Encoding is used with the Necklace System: The inner circle of the WhyCode marker encodes a binary string which is bit-shifted to match a Necklace code. Apart from identification, the number of bit-shifts allows us to identify the marker's rotation [5].

2.2 Systems used for bee tracking

Honey bees present a challenge to various multi-object detection and tracking systems. That is due to the high density of nearly identical specimens, constantly moving and occluding each other. Usually, these systems are accompanied by physical fiducial markers that are put on some or all members of the honey bee colony. Fiducial markers facilitate the detection of individual honey bees, but they can also get in the way of the bees' movement. Some systems do not require markers. Instead, they rely on extracting distinct visual features through the use of convolutional neural networks. This section will elaborate on both of these approaches.

2.2.1 Markerless tracking

One of the systems that use distinct visual features is the 'pixel personality' system. This system combines position and angle loss function, which allows for simultaneous computation of both bee orientation and class. The convolutional neural network (Fig. 2) distinguishes between a fully visible bee and a bee inside a comb. The output of this network is the x and y coordinates and the estimated angle and class of the honey bee [22].

From the output of the detection network shown above is calculated the similarity measure used for matching detections across neighbouring frames. Honey bee tracks became only detections that were sufficiently close, had low similarity measure, and occurred for at least 50 consecutive frames. These tracks were then used as a training dataset for learning pixel personalities. The training process consists of two repeating steps, learning,

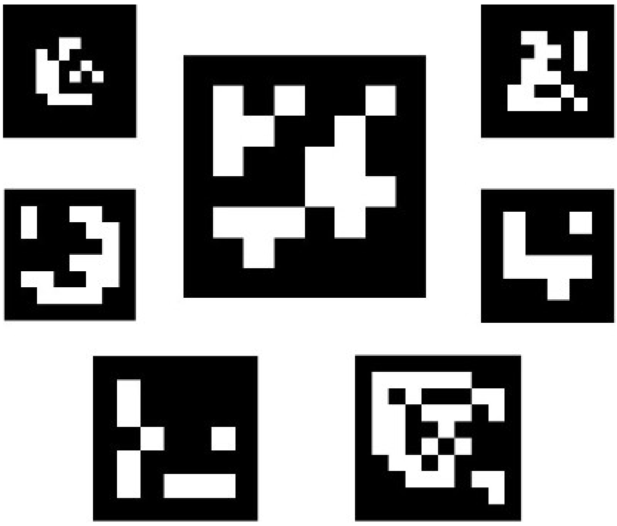


Figure 8: Possible ArUco markers.[6]

and matching. The network is trained until its loss reaches the selected threshold in the training step. Then in the matching step, the learned pixel personalities are used to match together the most similar detection in the following frames that were not a part of the training set, the closest detection is then added to the training set.

Another system that builds upon the pixel personality approach by using pixel personalities for its initial training dataset can additionally detect filled cell combs, see Figure 10, and thus keeping count of the brood numbers. This system uses a modified aforementioned neural network for detection that uses a 3-class softmax function that separates background, visible bee, and bee in a comb. Mainly, this extended system uses multiple visual features per bee, vector embeddings, rather than a singular pixel identity. Vector embeddings allow for an easy calculation of similarity between bee detections and matching corresponding detections together as a track. The tracking convolutional learns by comparing the correct matching of detections against an incorrect matching by measuring the Euclidean distance between the vector embeddings of the pairs.

The figure 10 shows the training process of the two neural networks. The training inputs for the bee detection network are a class segmentation map and an angle segmentation map. They are created from a hand-annotated image patch, yellow arrows for fully visible bees, and a yellow circle for bees inside a comb cell. An image segment is fed into the network and passed through its layers, each with its own number of convolutional filters. The network uses the data from the previous frame to enhance detection precision. The network produces two images, one with the bee class estimations and the other with the angle estimation. When combined together, the orange result is nearly identical to the hand-annotated image segment . The training input for the brood cell network is a manually annotated segmentation map of brood cells. An image segment is put into the network and passed through its layers, each with its own number of convolutional filters. An image with

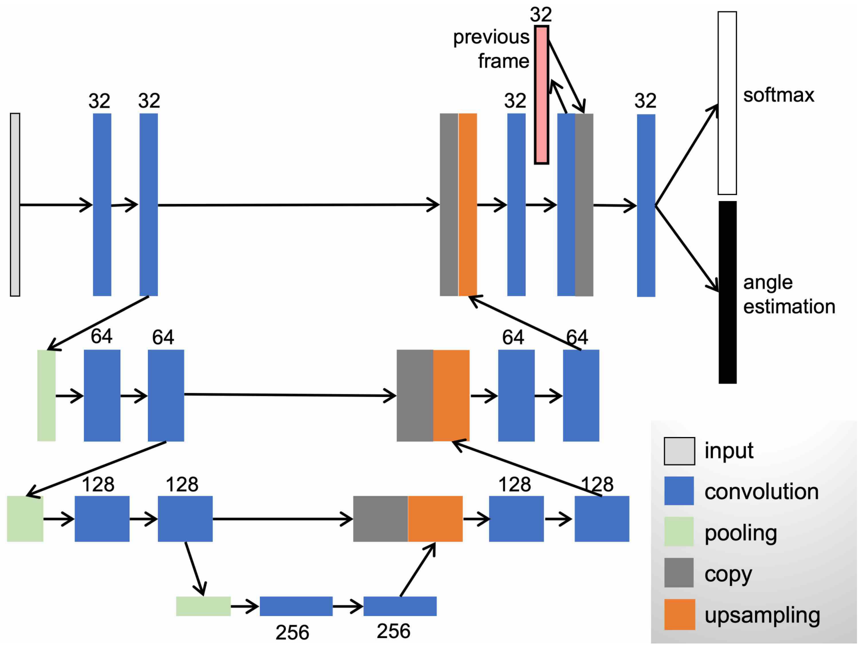


Figure 9: UNet architecture used for object detection. A temporal component was added (pink box) to incorporate information from the previous video frame as a prior to improve the prediction of object positions in the following frame. An additional convolutional layer was added before the nal two loss function layers - softmax for class prediction and angle loss function as proposed in [7].

position estimations of the brood cells is the output of the network.

This form of detection and tracking is relatively easy to use. One inhibition is that the dimensions of input images must be divisible by 256. Another big limitation is that the systems can not reliably differentiate between a worker bee and a queen bee.

2.2.2 Beesbook

Another approach to tracking the honey bee queen is to use markers. Beesbook uses a curved circular tag, 'Circulatrix', to best fit the bee's thorax, printed on a polyester film. The centre of the marker is divided into two halves, which shows the marker's and the bee's orientation. The white semicircle is always pointing towards the head of the bee. The marker is also capable of encoding 12 bit (4096) unique identification numbers [23].

The detection and decoding of the marker itself is done in multiple layers.

Preprocessor full honey cells are filtered out as they are brighter than the rest of the comb and may lead to false positives.

Tag localization regions with multiple strong edges are picked out and edges are extracted through Sobel filter. The edges are then eroded and dilated[24] to remove

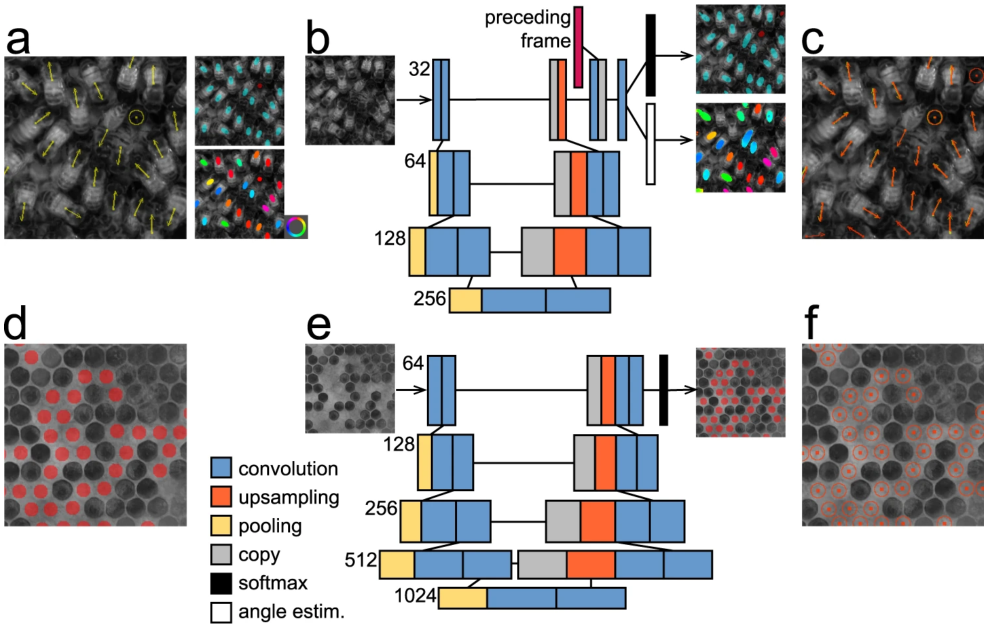


Figure 10: The diagram shows the working process of two convolutional neural networks, bee detection network(a,b,c) and brood cell detection network(d,e,f).

noise and connect neighbouring edges[23].

Ellipse fitting probabilistic Hough transform[25] is used to find ellipse-like configurations of pixels.

Grid fitting A 3D model of the marker('grid') is fitted, rotated in space, over every detected ellipse segment. If more than one grid configuration can be fitted to the ellipse then the best three are sent to the decoder layer.

Decoding Decoding of the identification number of the bee is done by computing the brightness of each segment of the middle ring and classifying the segments as either 0 or 1.

The Beesbook tracking system works in two steps. Firstly they detect bees on each frame and construct 'tracklets', which are trajectories of individual bees in consecutive frames. Secondly, they merge appropriate tracklets into one larger track of an individual bee.

In the process of creating a tracklet, each detection is given a group of possible candidates from the next frame. From each pair of the original detection and the candidate are extracted three features (euclidean distance, angular distance, Manhattan distance between their ID probabilities) that are then used to train a support vector machine. Detections that are not paired with an existing tracklet are the starting points for new tracklets.[3] For merging tracklets into tracks, the BeesBook system used six features as a metric to measure tracking results.

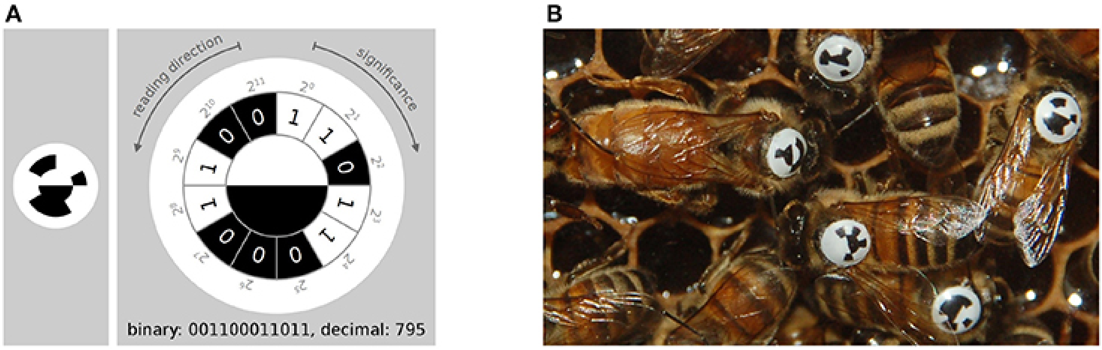


Figure 11: A - Visualization of the Circulatrix marker and its encoding. B - Circulatrix markers glued onto the bee's thoraxes

1. Manhattan distance of both tracklets' bitwise averaged IDs.
2. Euclidean distance of last detection of tracklet 1 to first detection of tracklet 2.
3. Forward error: Euclidean distance of linear extrapolation of last motion in first tracklet to first detection in second tracklet.
4. Backward error: Euclidean distance of linear extrapolation of first motion in second tracklet to last detection in first tracklet.
5. Angular difference of tag orientation between the last detection of the first tracklet and the first detection of the second tracklet.
6. Difference of confidence: All IDs in both tracklets are averaged with a bitwise median, we select the bit that is closest to 0.5 for each tracklet, calculate the absolute difference to 0.5 (the confidence) and compute the absolute difference of these two confidences.

These six were used in finding correct correspondences by a random forest classifier. The resulting tracks are of variable length and can contain gaps between frames with detection.

2.3 Selecting the most relevant method

To reliably detect the honey bee queen, we need a system that is capable of detection even in lower resolution in the observation hives and be accurate in differentiating the queen and the worker bees. The ArUco markers need to be quite large compared to the bees and are highly susceptible to occlusions. The Beesbook markers require to be spherical and high resolution of the processed frames. The markerless systems work well with lower resolution, but they can not reliably differentiate the queen from the worker bees. We have elected to use WhyCode because of its ability to detect the markers even on lower resolution[16], ability to detect in real-time, and the markers can easily be printed on paper.

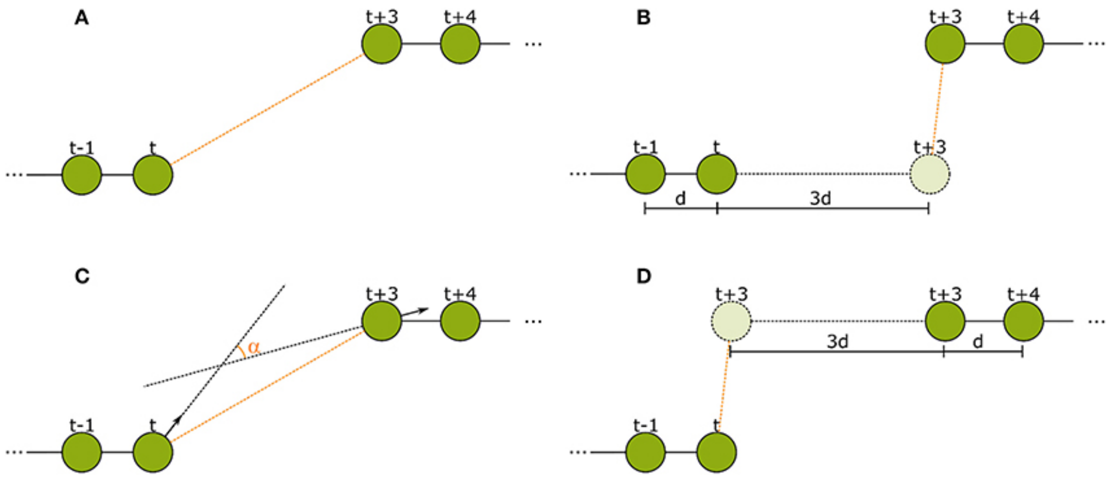


Figure 12: Visualization of aforementioned features. A - Euclidean distance, B - Forward error, C - Angular difference, D - Backward error

3 WhyComb

WhyComb is a proposed system for vision-based queen localisation. WhyComb builds upon the WhyCode system and specialises in detecting the honey bee queen. The WhyComb system implementation is spread over two Robot Operating System (ROS) nodes, a modified WhyCode node and a Cropper node, in addition to the WhyComb methods also crops the received image for further analysis of the queen's surroundings.

To enable a basic analysis of the honey bee queen's interactions with her surroundings, we integrated the WhyComb into a network of ROS nodes, see Figure 13. This graph shows the computation graph of our ROS nodes setup. Each observation hive has four recorded combs, one Collector node, Statistics node, and a Court detection node, which is described in more detail in [26]. In Figure 13 the hexagons represent the ROS nodes, and the arrows between them indicate the publisher (start of the arrow) and the subscriber of the topic (head of the arrow). Every observation hive and comb inside it has a unique prefix for its topics, so it is clear from which node a topic is being published. The orange highlighted ones, WhyCode and Cropper node, are where the implementation of the WhyComb is.

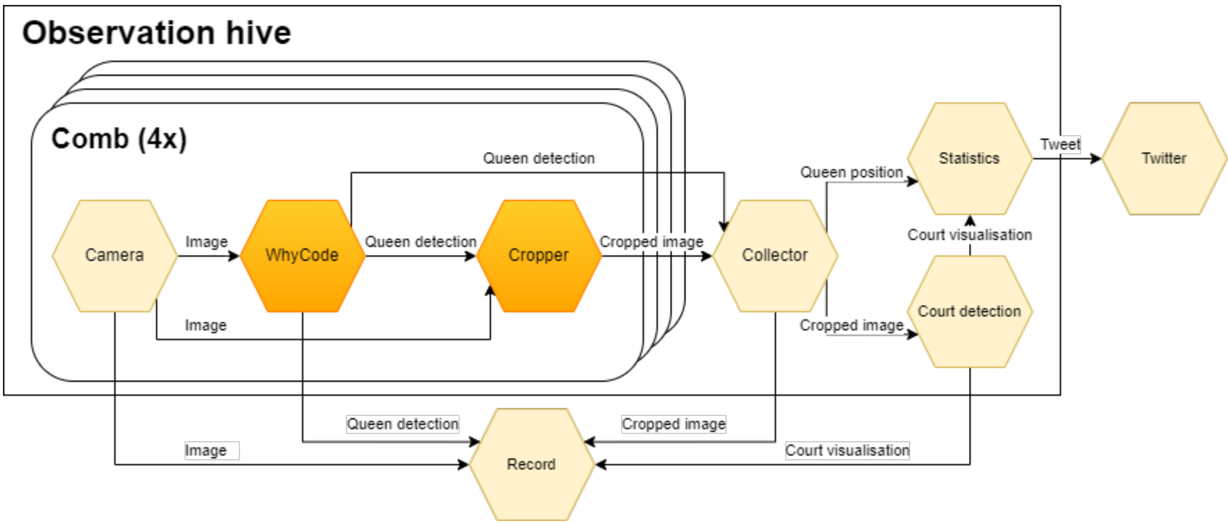


Figure 13: ROS computation graph of the observation setup in Graz.

Camera The Camera node is directly connected to the recording camera. It takes the camera's image output stream, converts it into a ROS image message and publishes the message in the Image topic.

WhyCode The WhyCode is subscribed to the Image topic, processes it in real time and publishes the detections in the Queen detection topic.

Cropper The Cropper node subscribes to the Image and Queen detection topics and based on their sequential numbers it matches the messages from both topics together. When the queen's marker is detected, it will use the WhyCode's detection coordinates to crop an area around the detection. If the marker is not detected, the Cropper will estimate the queen's position by convolutional pattern matching. The cropped area is published in the Cropped image topic.

Controller The Controller is in charge of filtering the Cropped image and Queen detection topics from each of the four combs in the observation hive. Based on the detections in the Queen detection topic it sends forward the crop that has the queen in it and her coordinates.

Statistics The statistics node gathers relevant information from the Queen position and court visualisation topics and publishes a summary of what happened in the beehive in the statistics topic.

Court visualisation The Court node detects worker bees around the queen on the cropped image. Then it uses support vector machine algorithm on the angle and proximity of the worker bees from the queen to identify the queen's court.

Record The Record node takes the subscribed topics and stores them into a ROSbag.

Twitter The Twitter node is publishing tweets based on the subscribed topic with statistics from the hive.

3.1 Filtering of false positive detections

One of the major weaknesses that we observed while working with WhyCode is the presence of false detections. The hive comb is made out of cells that are elliptical to hexagonal, which makes them susceptible to being mistaken for the queen's marker. This is due to the dynamically changing threshold for classifying black and white pixels. The comb cells are not the only elements liable to be falsely detected, and they can be any circular bits of debris or comb.

These false positives are interfering with our ability to track the queen reliably. If the queen's marker is obscured, and a false detection occurs on a different comb, the controller node will switch to it and gather data only from that specific comb.

The WhyCode checks for circularity, concentricity of the segment, and the number of pixels inside of the segment, but not for the size of the segment from the centre. We have extended the WhyCode code by adding an inequality equation of the general analytical ellipse equation (Equation 1) to verify that all the pixels inside the potential segment are inside an ellipse constructed from the eigenvalues and eigenvectors (Equation 2). WhyCode computes the eigenvalues and eigenvectors from the covariance matrix of the detected pixel segment. This allows for filtering out the falsely detected cells because they have a larger diameter than the queen's marker.

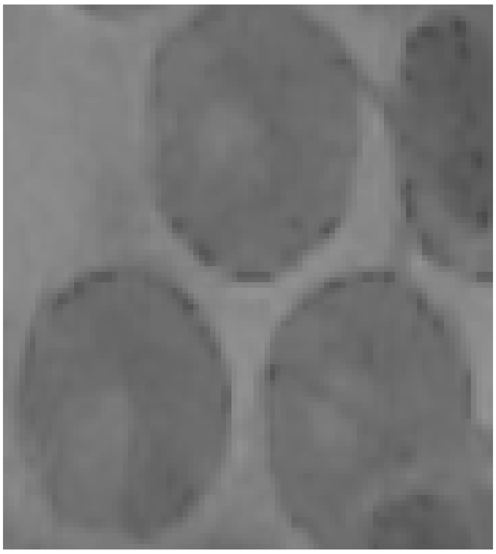


Figure 14: Falsely detected comb cell.

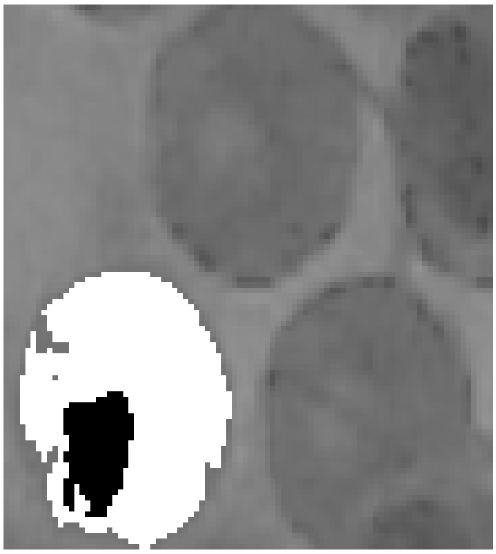


Figure 15: Falsely detected comb cell filled by WhyCode.

$$\frac{((u-u_c)\,cos(\alpha)+(v-v_c)\,sin(\alpha))^2}{a^2}+\frac{((u-u_c)\,sin(\alpha)+(v-v_c)\,cos(\alpha))^2}{b^2}=1 \quad (1)$$

$$((u-u_c)\,\nu_{1,u}+(v-v_c)\,\nu_{1,v})^2\,\mu_1+((u-u_c)\,\nu_{2,u}+(v-v_c)\,\nu_{2,v})^2\,\mu_2<d \quad (2)$$

In equation 1 is shown the general analytical equation for an ellipse. $u,\ v$ is the pair of coordinates in the Cartesian coordinate system of each point on the ellipse. $u_c,\ v_c$ are the coordinates of the centre of the ellipse. α is the angle by which the major axis is rotated. $a,\ b$ are the respective lengths of the semi-major axis and semi-minor axis.

Equation 2 is showing the analytical equation for ellipse with the use of eigenvalues and eigenvectors. ν_1 and ν_2 are the unit eigenvectors of the segment's covariance matrix, $\mu_{i,u}$, $\mu_{i,v}$ are the u and v coordinates of the i-th eigenvector, substituting the $cos(\alpha)$ and $sin(\alpha)$, and representing the rotation of the ellipse. $\mu_1,\ \mu_2$ are the eigenvalues of the matrix and are representing the lengths of the semi-major and semi-minor axis.

Another filter that we added to WhyComb to filter out the false detections is a size filter inside the Cropper node. The detections of debris and comb bits are much smaller than the marker. The filter also filters out the four calibration markers on the corners of each comb from the detection message received from the WhyCode node.

3.2 Dealing with occlusions

The second major problem we have encountered is the occlusions of the queen's marker by the worker bees. WhyCode is quite sensitive even to minor obstruction of the marker so that only part of it is hidden, but the rest is visible.

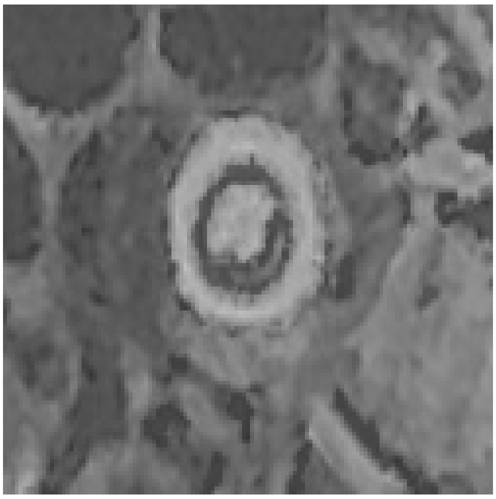


Figure 16: Detected WhyCode marker.

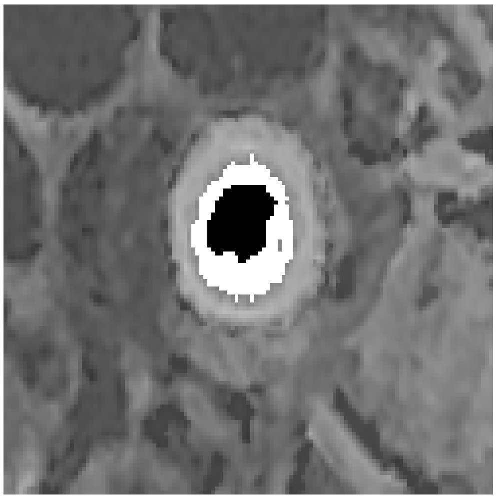
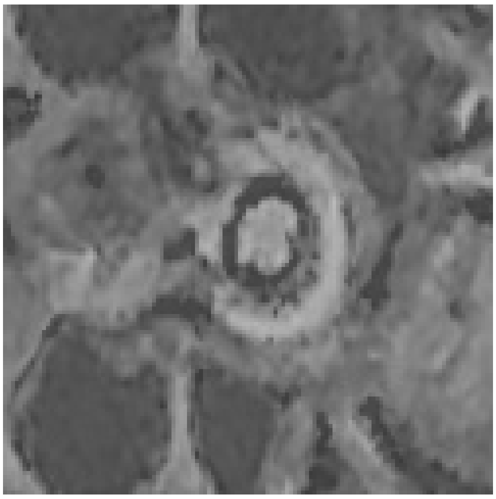
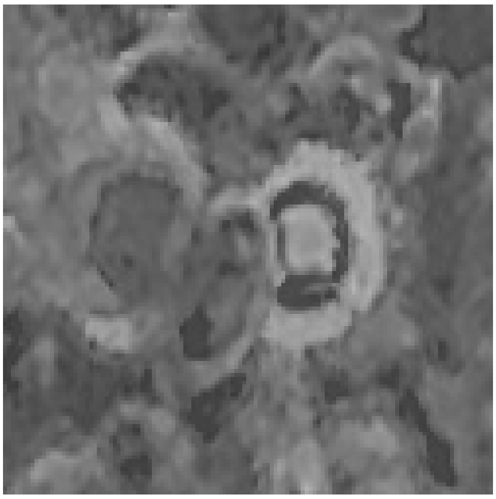


Figure 17: Detected WhyCode marker filled out by WhyCode.

Figure 18: Occluded markers





One of the possible solutions is to store the frames in which the queen was not detected and wait for the next detection. We linearly interpolate the queen's position from the two newest detections when the queen is detected. This is quite an easy solution to missing detections , but it does not guarantee the precise position of the queen if the queen moves in a nonlinear path.

Another possible solution is to publish the cropped images at the last detected coordinates. This way, it would be possible to process an image before the next one arrives and without the need to have the storage space for it if necessary, but it could happen that the queen would move out of the area that is being cropped, thus losing sight of her.

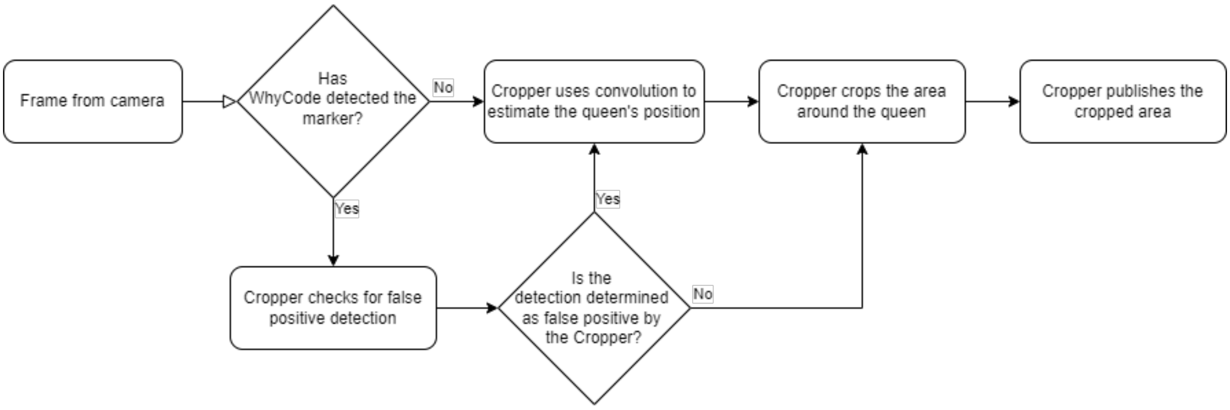


Figure 19: Graph of the interaction between WhyCode and Cropper in the ROS pipeline.

The solution that we have selected to be implemented in WhyComb is to substitute the WhyCode with a convolution when the detection fails. Convolution is a mathematical operation where a kernel, a matrix of weights, is moved over a picture. In each position, element-wise multiplication between the values of the kernel and the pixels under the kernel, the sum of this multiplication is then saved as an output entry. We have used a kernel modelled after the WhyCode marker, only with the necklace encoding zeroed out to account for the rotation of the queen on the comb. The convolution is done on a small area around the last WhyCode detection. Then from that area, the coordinates with the highest value are selected and used as the new detection.

If the queen's marker is completely obscured, the maximum from the convolution could move away from the queen because this solution cares only about the pixel with the highest value, even when the marker is not visible.

Figure 20: WhyCode marker. Red corners are not part of the marker, they define the outer white circle.

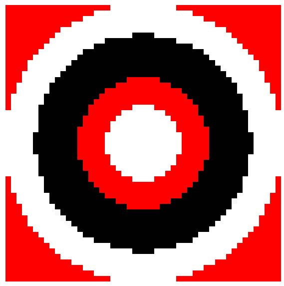


Figure 21: Convolution kernel based on the WhyCode marker. Red part is zeroed out to not affect the result of the convolution

4 Datasets

Our datasets were recorded at the Karl-Franzesns-Universität in Graz, Austria. The recording setup in Graz is a single observation hive 22 that holds four honeycombs. On both sides of the hive are two 12Mpx infrared cameras 23, each one connected to an NVIDIA Jetson Nano microcomputer. The Jetsons are running an Ubuntu operating system, and the Camera, WhyCode and Cropper nodes. On both sides of the cameras are infrared LED lights to provide the necessary illumination for observing the combs.

Figure 22: Observation hive from side with the bee tunnel.

The recording setup of ROS nodes can be seen in 13, the Recorder node runs on a separate computer and records relevant topics from the observation hive into one rosbag. Each recorded ROS topic has a comb and observation hive number, a unique prefix indicating its comb and hive of origin. One observation hive averages 50 GiB of data per day.

We used infrared cameras and lights to not disturb the honey bee colony, as the bees are incapable of perceiving it. The cameras' output is greyscale images with a width of 4032

Figure 23: Infrared cameras recording combs in the observation hive.

pixels and a height of 3040 pixels. The framerate of the recording cameras is six frames per second. The observation hive is covered in black cloth to prevent the daylight from seeping into the hive, thus imitating the darkness of a normal beehive 24.

Figure 24: Covered observation hive, to emulate the conditions inside a normal beehive.

The first dataset, 'False Positive dataset', was created by gathering frames where WhyCode had a false positive detection and contains 995 images. The filters in section 3.1 were tested on a subset of 75 images from this dataset.

The second dataset, 'Two Markers dataset', recorded soon after the recording setup was finished, contains 453 frames. As it was early after the marking of the queen, the worker bees tried to pry the marker from her. That is why in this dataset, there are two marked bees 25. If the workers succeeded in removing the marker, we could still track the other marked bee. This dataset was used to evaluate methods in section 3.2.

The third dataset, 'Occluded dataset', was recorded a week later, and contains 27403 frames. The worker bees have accepted the marker on the queen, so they treated it as a part of her body and frequently occluded it. Only a subset of 800 frames from this dataset was used in evaluating methods in section 3.2.

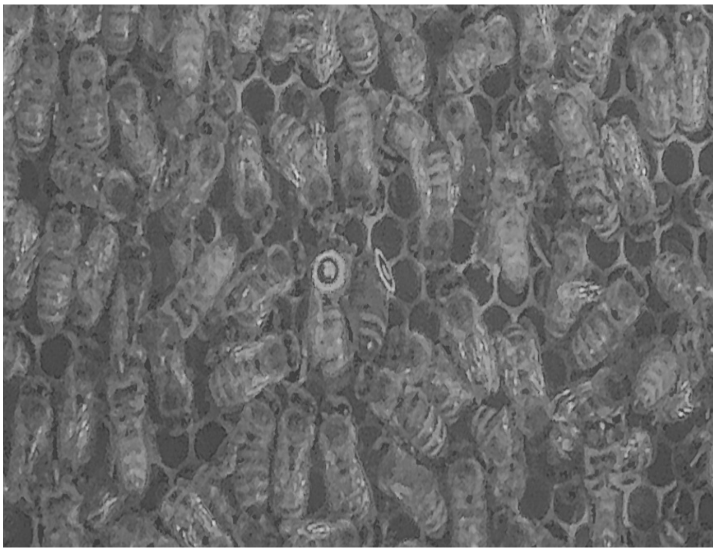


Figure 25: Queen with the other marked worker bee.

The datasets were annotated by hand in the Label Studio [27] annotation tool.

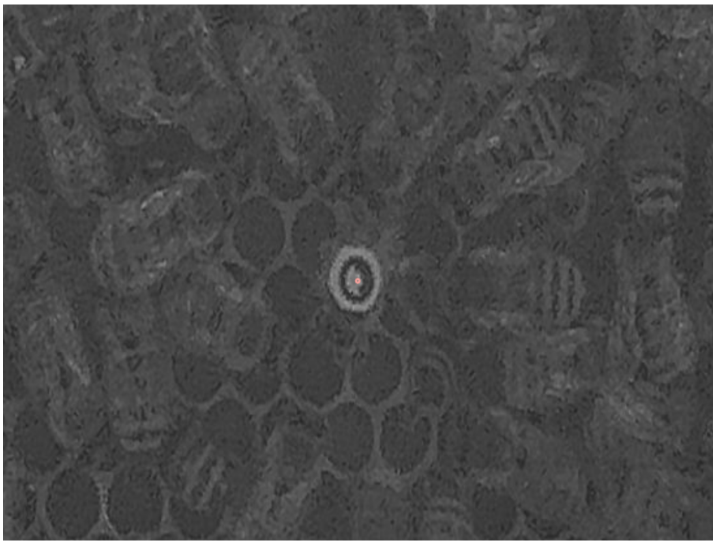


Figure 26: Annotated frame from the Occluded dataset in LabelStudio.

5 Experiments

We selected two methods, WhyCode and WhyComb, to evaluate their performance in localisation and detection. *Precision* and *Recall* were used to evaluate the detection of the methods. It is a standard performance metric for detection algorithms, but it neither reflects the magnitude of the error of the false detection nor how near the successful detection from the ground truth. To measure and compare the accuracy of the localisation we performed a Wilcoxon pairwise test over the calculated error vectors. To provide further insight, we also calculated the cumulative distribution function over the errors.

To calculate the error vectors for the experiments, we used the detections of the WhyCode, WhyComb, WhyCode, and just convolution, on the 'Two marker' and 'Occluded' datasets. Our form of representing a detection on a frame consists of two values, x and y coordinate together, and they describe any pixel in the frame. Because the WhyCode does have significantly fewer detections than the WhyComb, we used the coordinates from the last successful detection and used them in the following frames where WhyCode did not detect the queen. We calculated the Euclidean distance between its detection and the hand-annotated ground truth on all frames in both datasets for each method. This resulted in error vectors for all the methods, one vector per method per dataset. The error vectors are the primary data entry for the tests, even for precision and recall, as we can detect a false positive detection by thresholding the WhyComb method.

5.1 Precision and recall

While retrieving information, in our case, information about the detection of the queen, every retrievable item can be classified into one of four categories defined by the two characteristics: *Retrieved or Not Retrieved*, *Relevant or Not Relevant*. The four categories are *True positive*, retrieved and relevant, *False positive*, retrieved but not relevant, *True negative*, not relevant and not retrieved and *False negative*, relevant but not retrieved [28].

Precision is the purity of the retrieval of information and is the number of retrieved relevant items divided by the number of retrieved items 3. Recall is the effectiveness of including relevant items in the retrieval and is the number retrieved relevant items divided by the number of all relevant items that could be retrieved 4.

$$Precision = \frac{True\ positives}{True\ positives + False\ positives} \tag{3}$$

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives} \tag{4}$$

The WhyComb will always retrieve an estimation of the position of the honey bee queen due to the convolution pattern matching will always find a local maximum. We have selected a threshold of detection error that will separate the true positive detection from false-positive ones. It is the largest detection error that was produced by WhyCode detection, 5.1. We did so instead of using a precision and recall curve to make convolution consistent with WhyCode.

Method	Precision	Recall
WhyCode	100.0%	61.9%
WhyComb	99.8%	100.0%

Table 1: Precision and recall in Two Markers dataset

Method	Precision	Recall
WhyCode	100.0%	24.9%
WhyComb	98.5%	100.0%

Table 2: Precision and recall in Occluded dataset

Method	Precision	Recall
WhyCode	100.0%	38.3%
WhyComb	99.0%	100.0%

Table 3: Precision and recall in combined dataset of Two Markers dataset and Occluded dataset

5.2 2D error analysis

The aim of the 2D error analysis is to statistically and qualitatively compare the precision and error of each method. To conclude which method performs statistically significantly better, we used the Wilcoxon pairwise test on the pairs of error vectors created from the same dataset. For the qualitative comparison, we computed the cumulative distribution function of the error vectors on both datasets.

5.2.1 Statictical analisys

The Wilcoxon pairwise test is a nonparametric test used for two independent samples that do not have a Gaussian distribution. Its null hypothesis is that, for randomly selected values X and Y from two populations, the probability of X being greater than Y is equal to the probability of Y being greater than X. When the probabilities are equal, it means that the two populations have the same distribution [29]. We performed the test on the error vectors of WhyCode and the combination of WhyCode and convolution on both datasets.

From 4 and 5 we can see that the p-value of the Wilcoxon pairwise test is less than 0.001 for both datasets, so we reject the null hypothesis and can see that there is a statistically significant difference between errors of the two compared methods. Due to the statistical significance, we can compare the means of both methods' error vectors to see which performs better 6.

H_0	p-value
$Distribution_{WhyCode} = Distribution_{WhyComb}$	0.00

Table 4: Result of the Wilcoxon pairwise test on Two Markers dataset

H_0	p-value
$Distribution_{WhyCode} = Distribution_{WhyComb}$	0.00

Table 5: Result of the Wilcoxon pairwise test on Occluded dataset

Method	$\mu_{dataset1}$	$\mu_{dataset2}$
WhyCode	4.79	10.52
WhyComb	2.71	4.25

Table 6: Means of both methods on both datasets.

In 6 you can see that the mean of WhyComb is lower by a significant margin. Due to this fact, we compared the error vectors of WhyCode and only convolution, but only on the frames where WhyCode had a successful detection to see if there is any statistical difference between the errors of the true detections. Suppose there was a statistically significant difference, and the convolution performed better. In that case, we could modify WhyComb further, having WhyCode select the area where the marker is and pinpoint the location with convolutional pattern matching.

H_0	p-value
$Distribution_{WhyCode} = Distribution_{WhyComb}$	0.00

Table 7: Result of the Wilcoxon pairwise test on a partial Two Markers dataset

H_0	p-value
$Distribution_{WhyCode} = Distribution_{WhyComb}$	0.11

Table 8: Result of the Wilcoxon pairwise test on a partial Occluded dataset

From the results in 7 and 8 we cannot reject the null hypothesis. Both datasets are the same observation hive, just recorded at different times. That is why we decided to merge

H_0	p-value
$Distribution_{WhyCode} = Distribution_{WhyComb}$	0.18

Table 9: Result of the Wilcoxon pairwise test on a combined partial Two marker and Occluded dataset

Method	$\mu_{partial\ TwoMarkers}$	$\mu_{partial\ Occluded}$	$\mu_{combined\ partial}$
WhyCode	2.64	2.46	2.56
Convolution	2.70	2.23	2.51

Table 10: Means of both methods on all partial datasets.

Method	$\sigma_{partial\ TwoMarkers}$	$\sigma_{partial\ Occluded}$	$\sigma_{combined\ partial}$
WhyCode	0.67	0.62	0.66
Convolution	0.75	0.72	0.77

Table 11: Standard deviations of both methods on all partial datasets.

the two partial datasets to try to get more relevant results because as the number of cases increases, the result gets more precise. This test 9 showed us that we cannot reject the null hypothesis and that the means and standard deviation are quite similar **??**. From the results on the combined partial dataset, we decided not to prioritise convolution for detection in the combined method.

5.2.2 Qualitative analisys

Cumulative distribution function (F) of random variable X at x is the probability that X will be a value equal of less than x, shown in equation 5. This means that the steeper the curve of the function goes to 1, the smaller the standard deviation is, and also that the majority of measurements have a small value.

$$F_X(x) = P(X \leq x) \tag{5}$$

The cumulative distribution function's graphical representation offers a more intuitive comparison of the tested methods than the statistical analysis. For our evaluation, we used the calculated error vectors of each method to create the cumulative distribution error function. The graphs are made from the same error vectors that the Wilcoxon pairwise test 5.2 was conducted from.

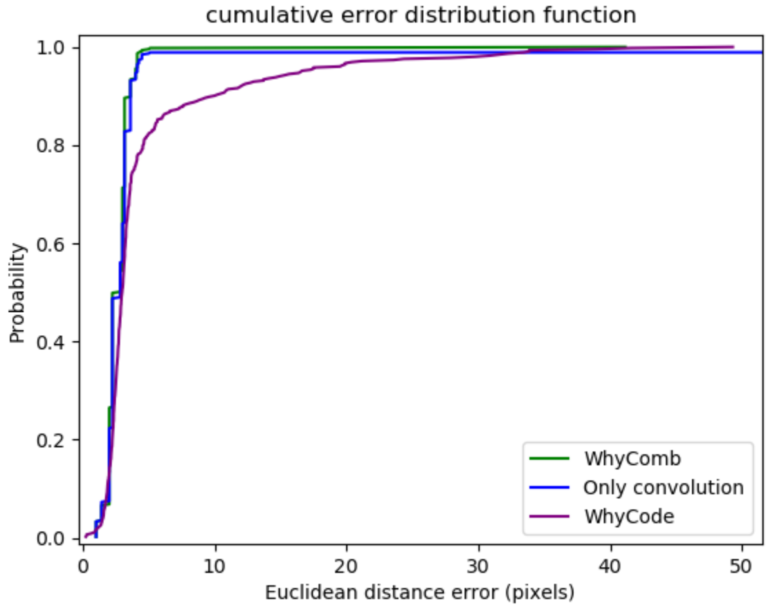


Figure 27: The graph depicts a cumulative distribution function errors of WhyCode, WhyComb and Convolution on the Two Markers dataset. The function shows the probability that the method will have error equal or less than the threshold on the x axis.

5.3 Filter testing

To evaluate the filters implemented in section 3.1, we used a subset of the False Positive dataset of 75 frames. The filters performed well, filtering out all the false detections resulting from this dataset. The size filter removed 68 of the false detections, while the ellipse filter discarded the remaining seven false detections.

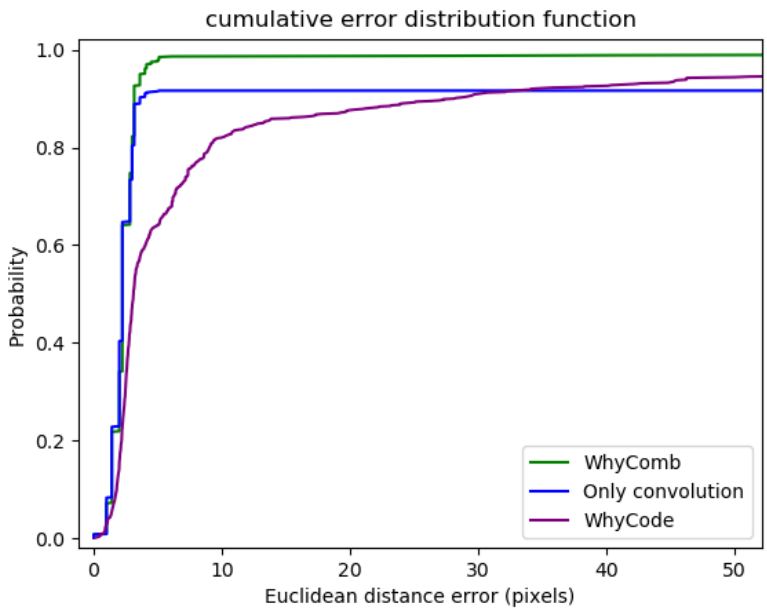


Figure 28: The graph depicts a cumulative distribution function errors of WhyCode, WhyComb and Convolution on the Occluded dataset. The function shows the probability that the method will have error equal or less than the threshold on the x axis.

6 Conclusion

This thesis aimed to implement a system capable of automatic detection and localisation of the honey bee queen. The system implemented in this thesis called WhyComb, is built upon the WhyCode system, which uses a circular black and white marker with a binary identification number encoded inside. The WhyCode is not suited for the working conditions of a observation hive due to the frequent occlusions of the marker and comb cell being similar to the marker.

WhyComb handles these problems by implementing two-factor filtering of detections and substituting WhyCode's misses with convolutional pattern matching. The first filter checks if the detections are an ellipse, and the second filter judges the detection based on their size. The convolutional substitution uses a kernel created from the WhyCode marker with the slight modification that the binary encoding is zeroed out so that the kernel is centrosymmetric and thus invariant to the queen's rotation on the comb.

We evaluated the WhyComb's solutions to the aforementioned problems, the filters on how efficient they are, and the convolutional pattern matching on the detection and localisation. The results of our experiments showed that the WhyComb performs significantly better in the set tests.

To allow for the analysis of the queen's surroundings, we integrated the WhyComb into a network of ROS nodes. In the network, WhyComb provides positions of the queen and an image cutout of her surroundings to perform further analysis.

We are gathering more data for future improvements to the WhyComb. With them, we plan to do more rigorous testing and analysis. We plan to implement thresholding for the convolutional detection so that it can identify when the queen is completely not visible. In addition, we plan to add an angle estimation method to the WhyComb.

References

[1] Candy Rowe. Receiver psychology and the evolution of multicomponent signals. *Animal behaviour*, 58(5):921–931, 1999.

[2] Stefanec et al. A minimally invasive approach towards “ecosystem hacking” with honeybees. *Frontiers in Robotics and AI*, 9, 2022.

[3] Franziska Boenisch, Benjamin Rosemann, Benjamin Wild, David Dormagen, Fernando Wario, and Tim Landgraf. Tracking all members of a honey bee colony over their lifetime using learned models of correspondence. *Frontiers in Robotics and AI*, 5, 2018.

[4] Aleksa Corovic, Velibor Ilic, Siniša Duric, Mališa Marijan, and Bogdan Pavkovic. The real-time detection of traffic participants using yolo algorithm. In *2018 26th Telecommunications Forum (TELFOR)*, pages 1–4, 2018.

[5] Peter Lightbody, Tomáš Krajník, and Marc Hanheide. A versatile high-performance visual fiducial marker detection system with scalable identity encoding. In *Proceedings of the symposium on applied computing*, pages 276–282, 2017.

[6] OpenCV. Aruco in opencv. `https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html`.

[7] Katarzyna Bozek, Laetitia Hebert, Alexander S. Mikheyev, and Greg J. Stephens. Towards dense object tracking in a 2d honeybee hive. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[8] Bryan Danforth. Bees. *Current biology*, 17(5):R156–R161, 2007.

[9] Madeleine Beekman and FLW Ratnieks. Long-range foraging by the honey-bee, apis mellifera l. *Functional Ecology*, 14(4):490–496, 2000.

[10] Christopher I Keeling, Keith N Slessor, Heather A Higo, and Mark L Winston. New components of the honey bee (apis mellifera l.) queen retinue pheromone. *Proceedings of the National Academy of Sciences*, 100(8):4486–4491, 2003.

[11] Norman E Gary. Chemical mating attractants in the queen honey bee. *Science*, 136(3518):773–774, 1962.

[12] Thomas D Seeley. *The wisdom of the hive: the social physiology of honey bee colonies.* Harvard University Press, 2009.

[13] Karl Von Frisch. The dance language and orientation of bees. In *The dance language and orientation of bees.* Harvard University Press, 2013.

[14] Sergios Karagiannakos. Localization and object detection with deep learning. *Towards data science*, 2019.

[15] Kristen Grauman and Bastian Leibe. Visual object recognition. *Synthesis lectures on artificial intelligence and machine learning*, 5(2):1–181, 2011.

[16] Tomáš Krajník, Matías Nitsche, Jan Faigl, Petr Vaněk, Martin Saska, Libor Přeučil, Tom Duckett, and Marta Mejail. A practical multirobot localization system. *Journal of Intelligent & Robotic Systems*, 2014.

[17] Keerthy Kusumam, Tomáš Krajník, Simon Pearson, Grzegorz Cielniak, and Tom Duckett. Can you pick a broccoli? 3d-vision based detection and localisation of broccoli heads in the field. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 646–651, 2016.

[18] Jiří Ulrich, Peter Lightbody, Aaron Weinstein, Filip Majer, and Tomáš Krajník. Whycode: Efficient and versatile fiducial localisation system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) - Late-breaking result.*, 2018.

[19] Jiří Ulrich, Ahmad Alsayed, Farshad Arvin, and Tomáš Krajník. Towards fast fiducial marker with full 6 dof pose estimation. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 723–730, 2022.

[20] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[21] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[22] Katarzyna Bozek, Laetitia Hebert, Alexander S. Mikheyev, and Greg J. Stephens. Pixel personality for dense object tracking in a 2d honeybee hive. *CoRR*, abs/1812.11797, 2018.

[23] Fernando Wario, Benjamin Wild, Margaret Couvillon, Raúl Rojas, and Tim Landgraf. Automatic methods for long-term tracking and the detection and decoding of communication dances in honeybees. *Frontiers in Ecology and Evolution*, 3, 2015.

[24] Jean Serra. Introduction to mathematical morphology. *Computer vision, graphics, and image processing*, 35(3):283–305, 1986.

[25] Yonghong Xie and Qiang Ji. A new efficient ellipse detection method. In *2002 International Conference on Pattern Recognition*, volume 2, pages 957–960 vol.2, 2002.

[26] Dominik Dvořáček. Automatic analysis of worker bee behavior in the vicinity of the honeybee queen. Bachelor thesis, Czech Technical University, Prague, Czech Republic, 5 2022.

[27] Heartex. Labelstudio. `https://labelstud.io`.

[28] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

[29] Daniel Dostál. Statistické metody v psychologii. *Olomouc. Filozofická fakulta*, 2016.

Appendix

OwnCLoud Content

In table 12 are listed names of all root directories on OwnCloud, containing all the attachements.

Directory name	Description
Two Marker dataset	Two Marker dataset
Occluded dataset	Subset of Occluded dataset
False Positive dataset	Subset of False Positive dataset
rr_msgs	ROS messages used for communicaton between nodes
rr_courdetector	ROS Cropper node
rr_whycode	ROS WhyCode node

Table 12: OwnCloud Content