Using Deep Learning for Non-Invasive Pollen Detection on Honey Bees

Nicholas Dykeman
Data Science Lab
ETH Zurich
dykemann@ethz.ch

Dominique Heyn
Data Science Lab
ETH Zurich
heynd@ethz.ch

Abstract

In this work, we present a completely non-invasive system for counting bee and pollen traffic in a beehive. This system is designed to be generalizable to various beehives, rather than being restricted to certain background and entrance types. In addition, we present a prototype cloud-based system that allows beekeepers to upload video footage, automatically processes this, and returns gathered statistics to the beekeepers. We examine the performance of our models, and compare it to human performance, which we in many cases match or even outperform.

1 Introduction

Honey bees are an essential part of the Earth's ecosystem. An estimated third of food pollination is due to them, and they produce over a million tons of honey each year. Bee colonies can contain up to 50'000 bees, and pollinate 4'000 square meters of fruit trees in the hive's vicinity (BBC (2019), Statista (2019)). The type of pollen that bees collect varies, and is dependent on the flora near the hive. It is not only useful to understand how much pollen is collected, as this is an indicator of the health of a hive and its population size, but knowing the type of pollen collected is helpful for those with allergies. The Varroa mite, a small parasite that feeds off bee larvae and spreads by traveling on a bee's back, is a threat to a hive's health, as it weakens and ultimately kills bees; see Figure 9 for some examples of Varroa. Classically, to estimate the hive's health and population, the beekeeper needs to open up the hive to look inside it, which is stressful for the bees.

Recent advances in computer vision and artificial intelligence have opened new avenues for tackling these issues. Several attempts at counting bees and detecting pollen have already been made, however thus far most of these included modifications to the entrance of a hive, e.g., by reducing space or by artificially constraining the lighting conditions, making it easier to detect bees. Our goal with this work is to develop an approach for counting bees and pollen in a completely non-invasive manner. We only require a video camera to be placed outside of the hive entrance. We want our results to be of use to as many beekeepers as possible, hence a requirement was that our models and methods are independent of the beehive. This brings additional challenges, as different hives will have different types of entrances, and different backgrounds. We show that, given enough labeled data, the issue of varying hive types can successfully be overcome.

Finally, in Section 4 we present a prototype implementation of a fully scalable architecture that makes it easy for new beekeepers to add their hives to our system, meaning that they can upload their gathered video footage to our system, which processes it and returns that gathered statistics to the beekeeper.

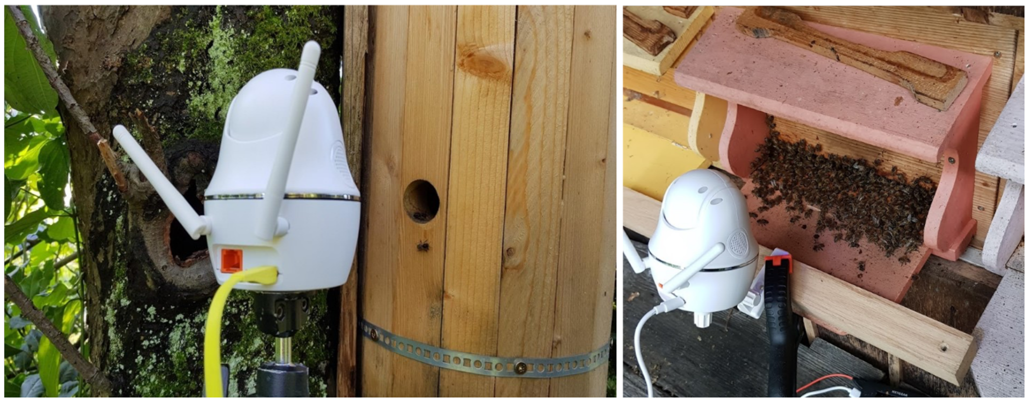


Figure 1: Non-invasive camera setup to record the bee videos.

2 Related Work

In recent years, there have been numerous approaches to detecting pollen on honey bees, as well as parasite detection (mainly the Varroa mite, Schurischuster et al. (2018)). To collect labeled data, Mégret et al. (2019) gathered video footage of the entrances of bee hives, which was then uploaded to a web based annotation system. On this platform, experts could label bees with tags such as "bearing pollen", as well as annotate different body parts of the bee. Rodriguez et al. (2018) and Sledevič (2018) have approached the task of pollen detection using convolutional neural networks (CNNs). Rodriguez et al. (2018) use a setup that helps ensure static lighting conditions, as well as using a spatially restricted entrance and blue background, making bees easier to detect. Their best model is a CNN trained on 710 images of bees, roughly half of which contain pollen. The images were all cropped to 180×300 pixels in size, and the bees were rotated such that they always faced upwards. The authors also tried using deep models, such as ResNet-50 (He et al. (2015)) and VGG16 (Simonyan and Zisserman (2014)). However their own, comparatively shallow model outperformed these. Sledevič (2018) uses a three layer CNN implemented on an FPGA. The model was trained with 2000 images in total, half of which contain pollen. It is particularly noteworthy that the images were taken using "natural" conditions, i.e., by simply placing a camera near the hive entrance, and from several different hives. The individual bee patches were extracted using a variety of methods, including background subtraction and color segmentation.

The work of Schurischuster et al. (2018) focuses on detecting the Varroa mite on honeybees. For this, the authors lead bees through small tunnels that fit only one bee at a time. By using transparent tunnels and artificial light, they recorded bees from multiple angles and were therefore able to detect parasites on body parts such as the underside of the abdomen, which are not visible in more natural setups.

3 Our Approach

The goal of our approach is to have a model that works as a complete pipeline, using recorded bee videos as input and providing beekeepers with data regarding bee traffic and pollen count. This objective underlies the constraint that our model should be general, in particular independent of specific lighting conditions and easily generalizable to new hives. At the same time, the data should be gathered in a completely non-invasive way. As such, we refrain from using artificial lighting and from spatially constraining the entrance to the hive. Data is collected using high-resolution (usually 2560×1920 pixels at 25 FPS) video cameras placed near the hive entrance. See Figure 1 for an example setup. To count the pollen traffic in the hive, the frames of the video are first processed by our object detection model to segment individual bees, followed by our pollen detection model to classify whether an individual bee is carrying pollen or not. We then consider short segments of five images around the time when bees are entering or leaving the hive. Each of these five bee images is classified as showing a bee with or without pollen, and the final decision is done using a majority voting scheme. This reduces the impact of misclassifications of the pollen classifier.

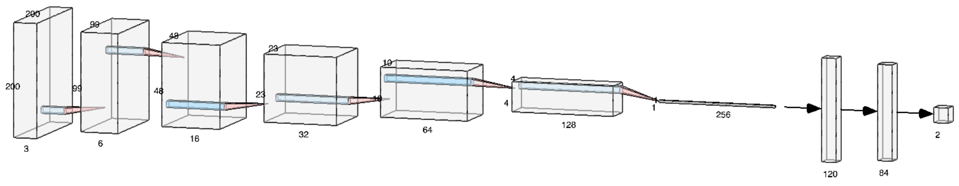


Figure 2: Architecture of the pollen classifier. All filters are of size 3×3, and the convolutional layers have $3, 6, 16, 32, 64, 128$ and 256 channels, and are of square sizes $200, 99, 48, 23, 10, 4, 1$ (graphic visualized using Lenail (2019)).

3.1 Bee Segmentation

The first step in our system is to detect individual bees in a given frame of the video. To this end, we used the TensorFlow Object Detection API (Huang et al. (2016)) in combination with the Faster-RCNN model (Ren et al. (2015)). Given videos of bees from multiple hives, we sampled frames from these at random, and annotated bounding boxes for the bees using Labelbox (Labelbox (2019)). Using the Leonhard Scientific Compute Cluster at ETH Zurich ("Leonhard" hereinafter), we trained the object detection model. After training, we exported the model into a frozen graph, which at inference time can simply be loaded, and then fed with images to produce bounding box predictions. We accept a proposed bounding box if the confidence of the network is above 80%.

3.2 Pollen Detection

To decide whether an individual bee carries pollen or not, we trained a classification network on images of individual bees. The data consists of image sections of 200×200 pixels that were labeled as containing a bee by the segmentation model, and classified these into pollen bearing, and non-pollen bearing bees. This model is a neural network with six convolutional layers, followed by three fully connected layers (see Figure 2). After each convolutional layer, a maxpooling layer is used, and as a last step the result of the last fully connected layer is fed into a softmax layer. This model is implemented using PyTorch Version 1.3.0, and trained on Leonhard.

3.3 Counting Pollen

One of the goals of this work was to count both the bee and pollen traffic entering and leaving the hive. Given a video of the bees, our algorithm proceeds as follows. For each frame of the video, the bees are located using the bounding box detector described above. It then compares these newly found boxes with the ones from the previous frame. We match the corresponding boxes between frames using the IoU score, also known as Jaccard index, as our metric. In other words, if two boxes from consecutive frames have the highest IoU score among all possible combinations, we assume that the bee is the same. If for a given box we cannot find a corresponding one in the previous frame, we consider that bee to be a "new" one. This occurs for example when a bee flies into the frame, or leaves the hive. If a box in the previous frame does not have a matching successor, we consider it "lost". This situation happens for instance when a bee enters the hive. This allows us to track the bees' trajectories as they enter and leave the hive.

To be able to decide if a bee entered or left the hive, we specify the hive entrance as a region of pixels in the image. Once we lose track of a bee, we process the positions at which it was detected. We discard bees that were neither found nor lost near the hive entrance, or that were both found and lost near the hive entrance, as these do not contribute to the net bee traffic. If, on the other hand, we either find the bee at the hive entrance or lose it there, we assume that the bee has either left or entered the hive, respectively. The individual bee images, generated by the bounding box network, are processed by the pollen detection network. This yields a label for each bee that either reads "pollen" or "no-pollen". If more than half of the images in the sequence are labeled as pollen, we assume that the bee was carrying pollen. Under the assumption that subsequent classifications of the pollen classifier are uncorrelated to some extent, this increases the total accuracy of predictions.

Oftentimes, bees do not enter the hive immediately but sit and move around near the hive entrance. Such bees are tracked over many frames, which can lead the classifier to confuse bees, in particular if one walks over another. A possible way to work around this is to consider the direction in which the bee is facing, and to only consider candidate bee boxes in the successor frame that contain bees facing in a similar direction. However, achieving accurate predictions of the direction the bee is facing is challenging, particularly in cases where bees are on top of each other, or bunched together. The main problem is that a given bounding box will in such cases often contain more than one bee, or parts of different bees, making the direction that the bee is facing ambiguous.

Our solution to this problem is based on the fact that when we consider shorter sequences of images, the chance that a bee is mistaken for another is smaller. As we are only interested in the net traffic of bees and pollen entering and leaving the hive, it is not necessary to track bees over long periods of time. Instead, we break up sequences of bees into sets of five consecutive images. Given such a set, we consider the start and end location of a bee as described above, and update the traffic accordingly. With this strategy, the net traffic should not be affected even if a bee walks around the entrance of the hive for a while before entering for good.

4 Infrastructure

The task of measuring bee traffic and counting pollen is of importance to many beekeepers. Thus, an objective of this project was to make it simple for new beekeepers to add videos of their own beehive to our system, such that they can also receive data regarding their own hive. Due to the limited scope of this project, a complete operational implementation was not feasible. However, we did produce a working prototype that incorporates most features described in the following paragraphs, with a clear outline of the features that are to be added.

Our proposed system includes two cloud platforms for labeling and processing data. For labeling, we use Labelbox (Labelbox (2019)), which provides a clean tool for creating labels such as bounding boxes and classifications. Furthermore, there is a complete API available. This enables us to automate the entire process, such as adding new labelers to the platform, uploading data, and downloading labels to re-train the classifiers.

For data storage and processing, we chose Microsoft Azure as our cloud compute platform. Due to the large quantity of data and variable processing requirements, the usage of a cloud platform is essential to our pipeline. A 30 minute high resolution video can take up to 2 GB of storage, and the quantity of beekeepers using our system, as well as how much video data they provide can change. To use our system, beekeepers simply need to record videos of the hive entrance. We provide each beekeeper with a unique access token and endpoint URI, to which they can upload the videos. These videos are then stored in Azure Blob Storage, and subsequently found and processed by our pollen counting algorithm, which runs on virtual machines. Once processed, the video is archived on Azure Blob Storage with a lower storage tier.[1] The reason the videos are not deleted is to allow potential further research, experiments and verification in the future. Furthermore the low-frequency access tiers on Azure storage are comparatively cheap. The results found by the pollen counting algorithm are stored and made available to the beekeeper for analysis. A visualization tool is a desired future extension to this work, but was out of the scope of this semester project. A schematic of our design is depicted in Figure 3.

We constructed a prototype software that lets an administrator add new beekeepers, as well as new virtual machines for existing beekeepers with a few simple button presses. This program connects to a master VM on Azure, which in turn takes care of creation and deletion of resources. The transfer of access keys and management of access to storage containers is also coordinated by this master VM, ensuring that each beekeeper only has access to storage containers that are assigned to their hives.

Once this pipeline is used by beekeepers, and a group of people labels data on Labelbox, the system will be set up to periodically check if new labels were added. If enough new ones are available, it will start a compute job to re-train the models. For our experiments (see Section 5), we manually triggered training processes and use Leonhard. Each training job was done using a single Nvidia GeForce GTX 1080 (Ti) accelerator. Training the bounding box classifier usually completed in less

[1] `https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers`

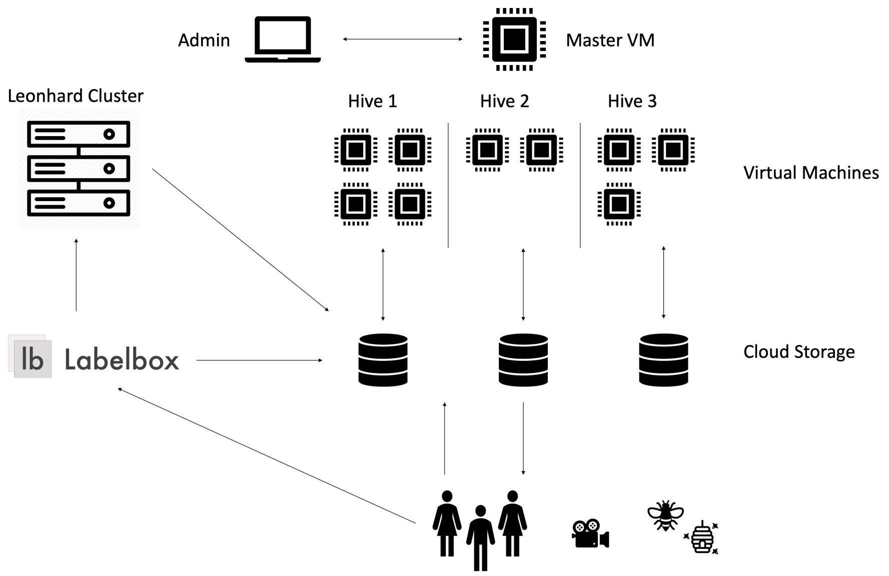


Figure 3: Our architecture, including Labelbox, Azure and Leonhard. Beekeepers upload video footage to the cloud. Annotators can label data on Labelbox; these labels are then periodically used to re-train the classifier. In our case this computation was performed on Leonhard. Virtual machines on Azure process the uploaded videos, and provide the beekeepers with information about their hive based on the videos they uploaded. The administrator can modify the system (e.g., by adding resources) via a master VM.

than four hours, however the training time depended a lot on whether a model is fully trained or partially re-trained. The pollen classifier was usually fully trained within a couple of minutes.

5 Models and Performance

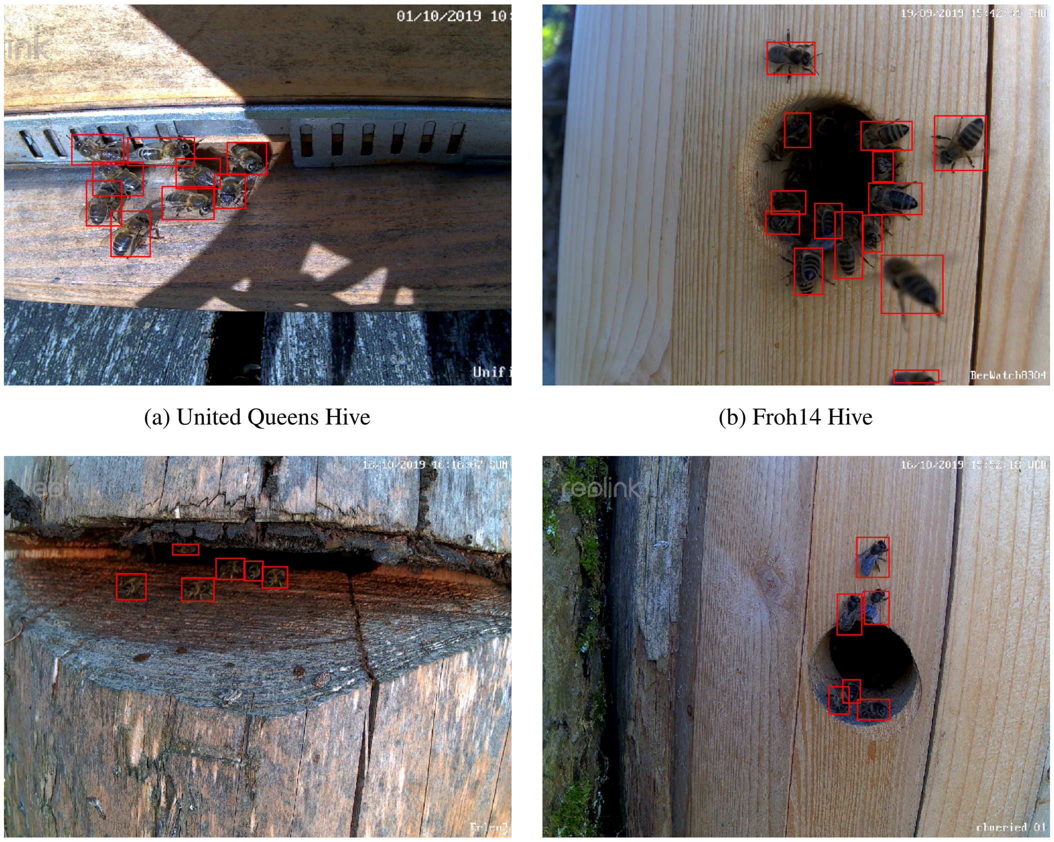
For our evaluation, we considered four different beehives (called *United Queens*, *Froh14*, *Froh23* and *Chueried*). We evaluated the performances of the bounding box and pollen classifiers, and how well out pipeline counts pollen in a video clip. All numbers are rounded to three significant figures.

5.1 Bounding Boxes

The performance metric we chose to evaluate the quality of the predicted bounding boxes for the bees is the IoU score. This is calculated by dividing the intersection of the ground truth and predicted bounding box by their union. There are however cases where the number of ground truth boxes is different from the number of predicted boxes, or it being unclear which ground truth box corresponds to which predicted box. This can occur due to misclassifications, and due to bees being on top of each other. We addressed this issue by considering it as a maximum-weight matching problem in a bipartite graph, where the two disjoint sets are the ground truth boxes and the predicted boxes, respectively. The IoU score of two boxes yields the weights of the edges connecting the boxes. Boxes that do not overlap have an IoU score of zero, hence to not produce an edge in the graph. In our experiments, this resulted in each node having at most four edges. This effectively means that the number of edges is $\mathcal{O}(V)$, where V is the number of nodes in the graph, in this case the number of bounding boxes.

For each of the four hives, we randomly sampled 150 frames containing bees, and labeled these. We did not include frames that either did not contain any bees, or that were taken at night. For the *United Queens* hive, we labeled an additional 300 frames for a further experiment. Figure 4 shows examples of images of such frames with bounding boxes for the different hives.

Our first experiment was to train a model on a single hive with 150 images and test the performance on all hives. We used 80% of the data for training and 20% for validation. Due to the variability


(a) United Queens Hive

(b) Froh14 Hive

(c) Froh23 Hive

(d) Chueried Hive

Figure 4: Examples of bounding boxes given by our model for each hive.

Table 1: For each hive, we trained a model on 150 images and tested it on all hives. 20% of the data was kept for validation purposes. The reported values are the averaged IoU scores of the bounding boxes. While the performance is fairly good on the hive on which the model was trained, generalization is poor.

Model \ Hive	United Queens	Froh14	Froh23	Chueried
United Queens	0.74	0.321	0.163	0.221
Froh14	0.466	0.637	0.229	0.315
Froh23	0.36	0.413	0.533	0.431
Chueried	0.148	0.329	0.2	0.589

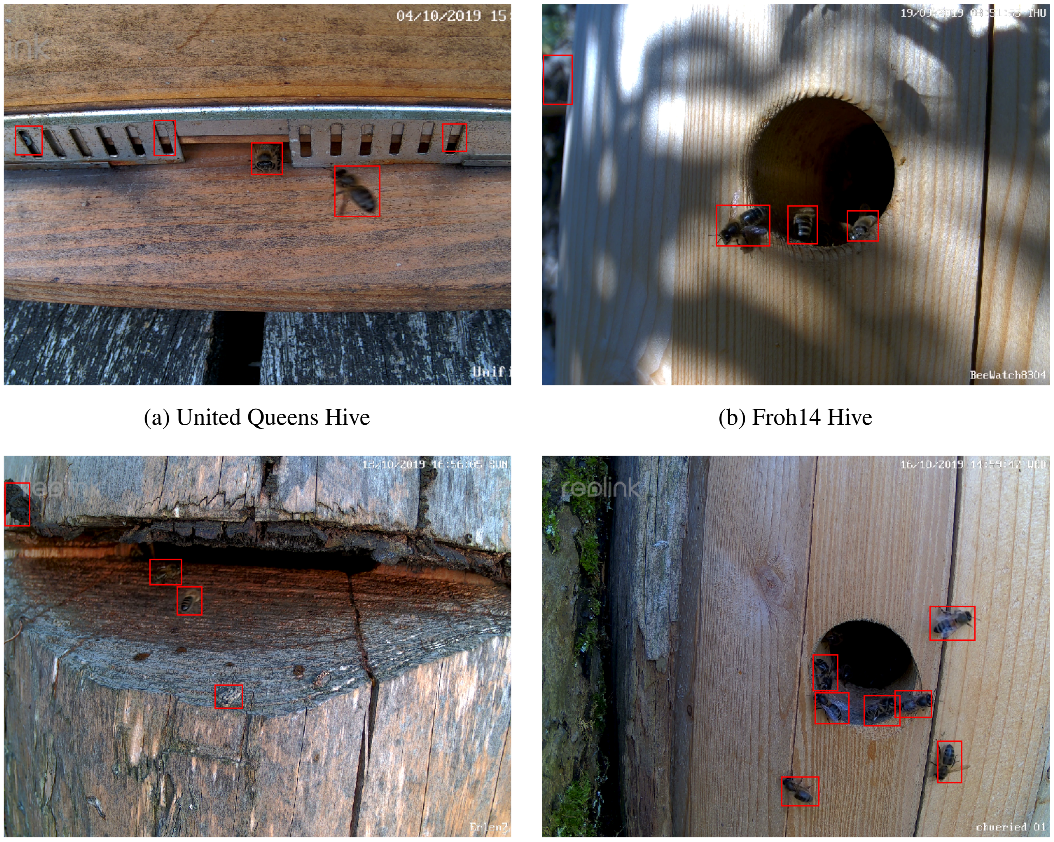
of backgrounds and fairly low number of training samples, we did not expect good generalization properties. The results can be seen in Table 1.

Next, we analyzed the transferability of the models. To this end, for each of the four hives, we trained a model using the images of the three other hives, and then tested the performance on all hives. Thus, each model was trained on 360 images. The results can be seen in Table 2. The comparison with the results in Table 1 shows that the performance of the models that were trained in this fashion on a given hive perform better than the models that were only trained on that hive. This shows that the models benefit not only from seeing images of one hive, but can benefit from others too.

It is helpful to not only consider the raw numbers, but to have a look at the predictions made by the models. Some example bounding box predictions are shown in Figure 5. It can be seen that the models make essentially zero false negative predictions, and that most mistakes arise from mistaking background artifacts for bees (nicely visible in Images 7a and 5c). Furthermore, bounding boxes that

Table 2: For each hive, we trained a model using the images of the three other hives, and evaluated it on all four. Here, ¬ means that we consider all hives but the one denoted.

Model \ Hive	United Queens	Froh14	Froh23	Chueried
¬ United Queens	0.252	0.81	0.632	0.724
¬ Froh14	0.807	0.401	0.674	0.607
¬ Froh23	0.816	0.806	0.32	0.737
¬ Chueried	0.785	0.792	0.665	0.416



(a) United Queens Hive

(b) Froh14 Hive

(c) Froh23 Hive

(d) Chueried Hive

Figure 5: Results of models trained on the three other hives, respectively.

are either too small or large, or not perfectly centered on a bee lead to low scores. For counting pollen traffic, it turns out that this is not as big an issue as one might think. The algorithm for counting bee traffic only considers bees that either start away from the hive entrance and end there, or vice versa. Given that artifacts in the hive background do not move, they will never contribute to bee traffic. The lack of false negative samples means that most of the bee traffic should still be counted correctly. A possible issue is that the bounding boxes do not always contain the full bee, meaning that pollen could in some cases not be captured.

Finally, we trained a model using all the images of all four hives together. This allows us to easily see the effect that adding the labels of the left-out hive from the previous four models has (see Table 3). The numbers show that when training models, it is crucial to include some images from the hive on which the model is intended to operate. It is relieving to see that the number of images required is fairly small. In our case we used 150 images, of which 120 were used for training. If multiple labelers contribute, this amount is easily achieved.

Table 3: The first row of this table shows the performance of the model trained on images from all four hives. The second row contains the diagonal entries from Table 2, i.e., models trained only on the other three hives.

Model \ Hive	U.Q.	Froh14	Froh23	Chueried
All Hives	0.799	0.77	0.673	0.635
Hive Left Out	0.252	0.401	0.32	0.416

Table 4: The IoU scores of the authors' labeled bounding boxes.

Hive	IoU Score
U.Q.	0.803
Froh14	0.676
Froh23	0.617
Chueried	0.543

So far, per hive, we only considered models trained on 150 images. To measure the performance gain by using more labeled images, we labeled an additional 300 images for the *United Queens* hive, and trained a model using a total of 450 images. As before, 80% of the images were used for training, 20 % for validation. The results are shown in Table 5. The performance is in both cases fairly poor on other hives, but this is to be expected as both models were only shown images with the *United Queens* background. The additional training does improve the accuracy at least in some cases, for an example, see Figure 7. Also, the performance on the hive that it was trained on improves notably. These results show that there is definitely a benefit to having more labeled images beyond just 150.

One reason that our models cannot achieve perfect accuracy is human error during labeling. Different labelers will make bounding boxes slightly different sizes, or have conflicting decisions on whether or not to label a bee that is barely visible in the hive entrance. Having more images labeled by multiple different labelers alleviates this problem, at least to some extent.

To illustrate this, the authors compared the bounding boxes that they created on several images, see Figure 6 and Table 4. Comparing these values to Table 3, we see that the IoU scores of the authors' boxes are lower than those achieved by the classifier, with the exception of *United Queens*, where the score is almost identical. This is due to two issues. Not only do different labelers draw bounding boxes of different sizes, as they for instance disagree on whether or not the legs should be included fully or not, but they also disagree to some extent for bees deep in the hive entrance if they should still be labeled as a bee or not. The classifier manages to achieve human-level performance, as it averages out human inconsistencies.

To conclude this subsection, we would like to point out one element of variability in the way we trained our models, which is not consistent across the four hives. While we did keep the number of images used for training and validation the same for all hives, the number of labeled bees within these images varies. We never labeled images without any bees, however we did not control for the number of bees in a frame. With the exception of the *Froh14* hive, the number of bees was fairly similar across the board. Table 6 shows the exact number of labeled bees in the training data.

5.2 Pollen Detection

Once the bounding box classifier detects a bee, a 200×200 pixel image is cropped out containing the bounding box, with scaling if necessary. These patches are then classified as showing pollen or not.

Table 5: Comparing the performance of the model trained on the United Queens hive using 150 and 450 images, respectively.

Model \ Hive	U.Q.	Froh14	Froh23	Chueried
U.Q. 150	0.74	0.321	0.163	0.221
U.Q. 450	0.819	0.345	0.202	0.297

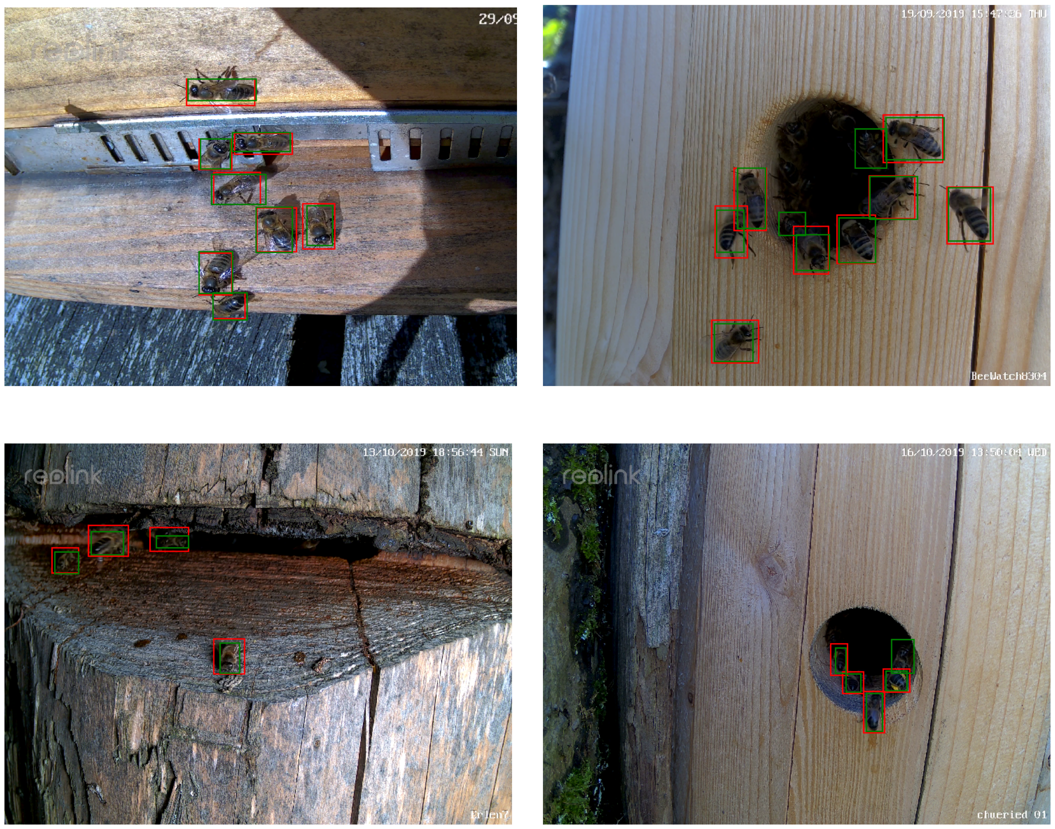
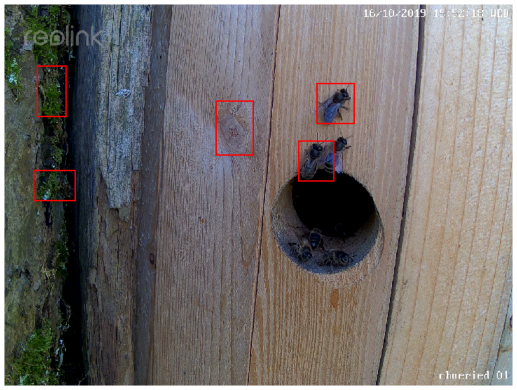


Figure 6: Examples of both authors' bounding boxes plotted for comparison. They not only vary in size, but also not always agree on what is still considered a visible bee.


(a) United Queens Hive, trained on 150 images, evaluated on the Chueried hive.

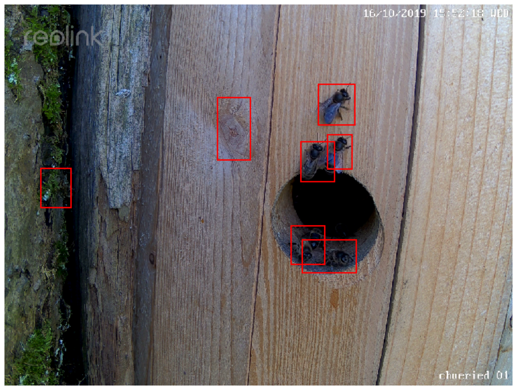

(b) United Queens Hive, trained on 450 images, evaluated on the Chueried hive.

Figure 7: Comparison of models trained on 150 and 450 images, respectively.

Table 6: The number of individual bees labeled for the training data (i.e., found in 120 images).

Hive	# Bees
U.Q.	616
Froh14	363
Froh23	510
Chueried	634

Table 7: The number of instances of the different classes for the hives.

Label \ Hive	U.Q.	Froh14	Froh23	Chueried
No Pollen	1232	45	50	54
Pollen	206	19	18	22
Total	1438	64	68	76

Table 8: Confusion matrices with true labels (rows) and predicted labels (columns). Here, P stands for the label pollen, and NP for no pollen. The classifiers were tested on the hives that are denoted in the top left entry, and trained on all other hives. The first table shows the result for the classifier that was trained on the hive United Queens (U. Q.) only, and tested on all other hives.

¬ U. Q.	P	NP
P	50	9
NP	15	134

Froh14	P	NP
P	15	4
NP	1	44

Froh23	P	NP
P	18	0
NP	5	45

Chueried	P	NP
P	18	4
NP	3	51

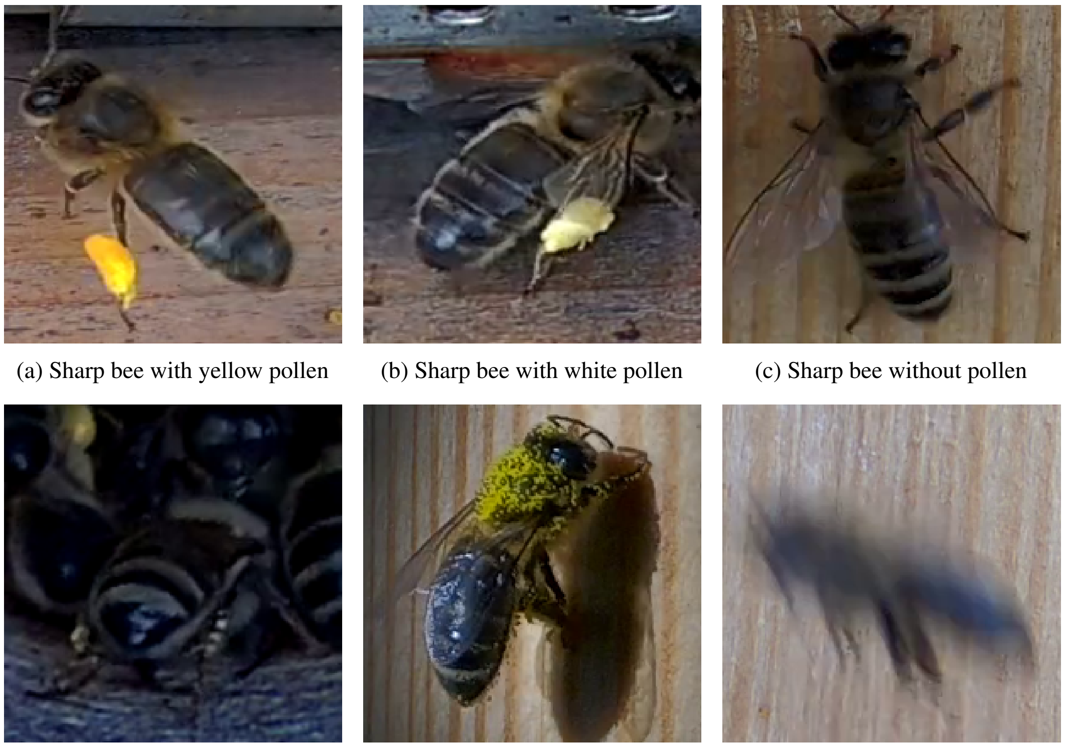
For this, we trained a pollen classification network using cross-entropy loss with adjusted weights to account for class imbalances. Since much of the analyzed footage was produced in late summer and fall, the number of pollen bearing bees is small compared to the number of bees that do not carry pollen. The number of instances per class and hive can be found in Table 7. For the hives *Froh14*, *Froh23* and *Chueried*, we used the bounding box classifier to produce the images that we labeled. This amounts to about 70 images per hive. For the hive *United Queens*, we labeled 1438 images total (see Table 7) with the goal of training the classifier on this hive, and subsequently adapting it by including data from additional hives. The rationale behind this approach is that the appearance of individual bees does not vary greatly between different hives. Therefore, the pollen detection classifier should be easily generalizable to perform well on new hives.

Figure 8 shows some of the challenges when training the classifier. First, the color of the pollen varies from white to yellow-orange. Once more video footage is collected throughout different seasons, this variation will further increase. Secondly, as depicted in Figure 8d, multiple bees can be present in one image, and thus pollen can be detected that is not carried by the bee that was found via image segmentation. Furthermore, bees sometimes carry pollen in atypical ways (see Figure 8e), or the image is blurry, making any classification difficult (see Figure 8f). Another challenge is the strong class imbalance. Initially, our classifier predicts every bee to carry no pollen, and achieves an accuracy of 70% and more. This has two implications. First, the confusion matrix should be analyzed, not only the total accuracy. Second, early stopping has to be used with caution and requires a long burn-in phase compared to the total number of steps. In most cases, the optimal result was reached within 100 steps.

Since rotation, scaling and translation of the images can move pollen out of the image, we did not perform pre-processing on the images. As our evaluation metric, we chose the F1 score. Table 9 shows precision, recall and the F1 score for the different classifiers. In the computation of the F1 score, predicting pollen was denoted as a positive result and predicting no pollen was denoted as a negative result. The table shows that when evaluated on a new hive, there is a slight improvement of training the classifier on multiple hives, compared to training it on one hive only. Two reasons for this are that adding more data generally leads to a better performance, and second that adding data from a new hive adds more variation in the data.

Table 9: Precision, recall and F1 score of the classifiers.

Score \ Hive	¬ U.Q.	Froh14	Froh23	Chueried
Precision	0.8	0.938	0.783	0.857
Recall	0.91	0.789	1	0.818
F1 Score	0.85	0.857	0.878	0.837



(a) Sharp bee with yellow pollen (b) Sharp bee with white pollen (c) Sharp bee without pollen

(d) Pollen on wrong bee (e) Bee with atypical pollen (f) Blurry bee without pollen

Figure 8: Examples of classifier training images

5.3 Counting Pollen

Due to time constraints on this project, this part was only briefly explored, and leaves much room for future improvement. The quality of our method at this time is not high enough yet that it can be used effectively, hence we only report on qualitative behavior and areas in which improvement is needed. The task of counting bees and pollen crucially relies on being able to first detect bees in frames, and then decide whether or not they contain pollen, hence our focus was predominantly on these two aspects. To be able to count traffic, detected bees need to be tracked over several frames in the video. An issue with our proposed algorithm is that it relies on two assumptions that are often not met, at least with certain hive types. Firstly, because we use IoU scores to match bees between frames, as described in Subsection 3.3, bees must move slowly enough so that their bounding boxes do not move too much. The second is that bees need to be isolated from each other to a certain extent, so that the maximal IoU score between boxes over two frames always match one bee, rather than two different ones. For the *United Queens* hive, the first assumption is mostly met, as bees must land before entering hive, due to the shape of the entrance. For the other three hives we considered, this is not the case. The round shape of the entrance makes it possible for bees to fly directly into the hive at high speeds, making is nearly impossible for our algorithm in its current implementation to detect them (it can even be difficult for humans watching the videos to spot the pollen). A honey bee has an average flying speed of 15 - 20 mph (UCANR (2013)), which translates to approximately 25 centimeters per frame, if the video is recorded at 25 FPS. Given that a typical bee is less than four centimeters long, this implies that bounding boxes of bees flying at these speeds will never overlap.

The issue of bees clustering together has two consequences. If a tracked bee suddenly "jumps" to another due to an mismatch, the position of the bee changes. In the case where one bee is entering and another is leaving the hive, it can result is a bee not being counted, as the bee that was being tracked entering (due to the mismatch) is now leaving, hence never enters the zone that we designated as the entrance of the hive. A second issue is that bees carrying pollen sometimes stand around in front of the entrance before entering. If other bees walk close by it, they can sometimes be classified to be carrying pollen, causing an over-counting of traffic. One would expect this phenomenon to

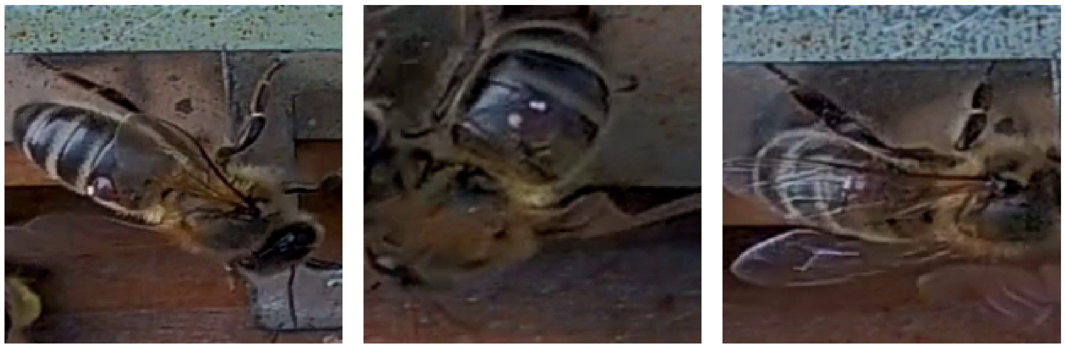


Figure 9: Varroa mites on Honeybees. Depending on the lighting conditions, the color of the mite is very similar to that of the bee (center), and mites can be covered by body parts of the bee (right). Note that these are instances where the mite is rather easy to spot, as in more difficult cases expert knowledge is needed.

occur symmetrically, i.e., to the same extent for bees entering and leaving the hive, thus balancing each other out. However, we observed the algorithm counting proportionally many more bees leaving than entering the hive, indicating that this effect is not actually symmetric.

An approach that could be investigated to remedy these issues is to consider the bees' orientation, as already mentioned in Subsection 3.3. It is sensible to assume that a the direction in which a bee is facing will not change very much between two frames. This would also work for bees flying at high speeds, as the direction in which they are flying typically will not change too drastically. Alternatively, a model could be built that, given a paired set of images, can match them to one another. Due to the fact that bees look fairly similar, and have moving features such as wings, we believe this latter approach is too challenging.

6 Outlook

This work lays the foundations for truly non-invasive and generalizable bee analysis. There are a number of avenues for further research available that can build upon our approach. As mentioned above, more work is needed to build a well-functioning bee tracking system. Incorporating the pose of a bee to estimate in which direction it is facing sounds promising. On some of the images containing individual bees, Varroa mites could be spotted. Counting these on bees entering and leaving the hive can be used to estimate the total number of parasites in the hive. This can help beekeepers to monitor the hives' health. However, detecting Varroa mites is more challenging than detecting pollen, since the color is less distinctive, and depending on the lighting conditions and their position on the bees' body, they are difficult to detect by non-experts (see Figure 9). Furthermore, there are generally very few Varroa mites. This makes the gathering of labeled data difficult.

As a second generalization, the color of the pollen can be analyzed. The color serves as an indicator from which plant the pollen originates. This can be used for allergy risk and biodiversity analysis. For this task, a segmentation of the image of individual bees is required to determine the exact position of the pollen, and a subsequent analysis of its color. Challenges in this task are varying lighting conditions, as these affect the apparent color of the pollen.

One point in which our work deviates from existing procedures is the non-invasiveness of our setup. This includes a natural structure of the beehive and in particular a background that is not specifically designed to facilitate the task of bee detection. Therefore, a background subtraction algorithm that is tailored for every new hive might be helpful in detecting bees. In particular, this could reduce the number of false detections that arise from spots in the background. We approached this by taking the median of all frames of a video, and subtracted this from the original video. However, this did not improve our results, since the bees sometimes have a similar color as the background, and because shadows and other dynamic obstacles led to false classifications.

The final goal of this work was to set up a prototype of a pipeline that enables beekeepers to record videos of their hive, and to receive an analysis of the number of bees and the amount of pollen that entered and left the hive. The next steps in realizing this project will be to obtain and label data from more hives, to transfer the models onto these hives, and to evaluate the pollen count using our models.

Acknowledgments

We would like to express our gratitude to Daniel Boschung, who was our primary supervisor for this project. Daniel is very passionate about bees and this project, and thus not only provided us with the opportunity to work on an interesting project, but also conveyed the importance to us. In particular, we would like to thank him for the honey he gave us that his bees produced. Furthermore, we would like to thank Professors Ce Zhang and Andreas Krause, who were the head supervisors of the Data Science Lab, as part of which this work was created. Their openness for assistance and discussions provided us with confidence and guidance throughout the project.

References

BBC. 2019. "Would we starve without bees, 2019. [Online]. Available: `https://www.bbc.co.uk/teach/teach/would-we-starve-without-bees/zkf292p`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:cs.CV/1512.03385

Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. 2016. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR* abs/1611.10012 (2016). arXiv:1611.10012 `http://arxiv.org/abs/1611.10012`

Labelbox. 2019. "Labelbox", Online, 2019. [Online]. Available: `https://labelbox.com`.

Alex Lenail. 2019. "NN-SVG", Online, 2019. [Online]. Available: `http://alexlenail.me/NN-SVG/AlexNet.html`.

Rémi Mégret, Ivan F. Rodriguez, Isada Claudio Ford, Edgar Acuña, Jose L. Agosto-Rivera, and Tugrul Giray. 2019. LabelBee: a web platform for large-scale semi-automated analysis of honeybee behavior from video. (2019).

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR* abs/1506.01497 (2015). arXiv:1506.01497 `http://arxiv.org/abs/1506.01497`

I. F. Rodriguez, R. Megret, E. Acuna, J. L. Agosto-Rivera, and T. Giray. 2018. Recognition of Pollen-Bearing Bees from Video Using Convolutional Neural Network. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 314–322. `https://doi.org/10.1109/WACV.2018.00041`

Stefan Schurischuster, Beatriz Remeseiro, Petia Radeva, and Martin Kampel. 2018. A Preliminary Study of Image Analysis for Parasite Detection on Honey Bees. In *Image Analysis and Recognition*, Aurélio Campilho, Fakhri Karray, and Bart ter Haar Romeny (Eds.). Springer International Publishing, Cham, 465–473.

Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556* (09 2014).

T. Sledevič. 2018. The Application of Convolutional Neural Network for Pollen Bearing Bee Classification. In *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. 1–4. `https://doi.org/10.1109/AIEEE.2018.8592464`

Statista. 2019. Honey market worldwide and in the U.S. - Statistics Facts. Available: `https://www.statista.com/topics/5090/honey-market-worldwide/`.

UCANR. 2013. "How fast can a honey bee fly?, 2013. [Online]. Available: `https://ucanr.edu/blogs/blogcore/postdetail.cfm?postnum=10898&`.