The Segmentation of Three-Dimensional Honeybee Swarm Images Using a U-Net

Nina van Hoorn[1], Danielle L. Chase[2], and Orit Peleg[3,4]

August 2024

1 Abstract

Honeybee swarms have been the subject of a multitude of research questions in the past, and though much has been learned about them, research has been limited by an inability to see inside the swarms. Though researchers have developed ways of estimating the density of certain areas within swarms, there is currently no way of knowing exactly where each individual bee is. This research seeks to develop a method of using a U-Net to segment bees in three-dimensional X-ray images of swarms. This network has proven to be capable of identifying individual bees and could help dramatically increase our understanding of honeybee swarms by allowing research to be done with knowledge of the exact location of each bee inside of these previously opaque super-organisms.

2 Introduction

Honeybee swarms are super-organisms composed of an aggregation of a large number of individual honeybees holding onto each other. They are temporarily formed when a group of bees, including a queen, leave an established hive to create a new hive and are found most often hanging from tree branches. Here, they face various changes in their external environment including temperature fluctuations, rain, wind, and shaking of the branch they may be clinging to. Previous research has found that the swarm as a whole efficiently adapts to these changes [2, 4, 5], but less is understood about the behavior of individual bees that allow for these changes to occur. There have been attempts in the past to gain more insight into the inner workings of the swarm by, for example, characterizing the spatially averaged mass of swarms [7], but the location of individual bees has never been characterized. Compared to dense aggregations

[01]Department of Computer Science, Skidmore College, email: nvanhoorn@skidmore.edu, [2]BioFrontiers Institute, University of Colorado, Boulder, USA, 80303; email: Danielle.Chase@colorado.edu, [3]BioFrontiers Institute, Department of Computer Science, Department of Physics, Department of Applied Math, Department of Ecology and Evolutionary Biology, University of Colorado, Boulder, USA, 80303 email: Orit.Peleg@colorado.edu, [4]Santa Fe Institute, Santa Fe, USA, 87501

of multicellular organisms, aggregations of single-celled organisms have been studied with single-cell resolution, due to specialized techniques in molecular biology and advanced microscopy. Deep learning methods have also proved to be capable of the segmentation of individual cells [1, 3, 6], allowing for a better understanding of how the individual cells make up these aggregations. The goal of this research is to explore the possibility of using the U-Net architecture found to be effective in image segmentation at the unicellular level for the segmentation of X-ray images of bee swarms. By doing so, we hope to better understand how the behavior of individual bees contributes to the overall structure of the swarm and allows for its adaptations in response to environmental changes.

3 Methods

To train a neural network capable of identifying individual bees in a swarm, training data is vital. First, swarms of bees were made and captured in three dimensions using X-ray Computed Tomography; this process is explained in-depth in "Collective Mechanical Adaptation of Honeybee Swarms" [4]. It is important to note that the data is obtained over a time span of roughly one minute; during this time the bees being captured can move, resulting in blurry images and large amounts of "noise" (Fig. 1). These noisy images are harder to segment for humans and neural networks alike.

Once a three-dimensional X-ray image of a swarm is obtained, it can be labeled to use for training data (Fig. 2). To label these images, the X-ray files were uploaded to *3D Slicer*, and individual bees were identified and colored in by hand as segments. As can be seen in figure 1, bees in these images show up as three individual components: the head, thorax, and abdomen. The connection between these three segments can be extremely narrow (at times, not even visible) and are only seen in certain slices. When labeling the individual bees, it was ensured that all three segments of the bees were connected in at least two slices, even if this meant artificially adding a connection where one was not visible. Additionally, the bees are often packed tightly together, also shown in figure 1, and, when labeling them, it can be challenging to differentiate the boundaries of the bees, especially in images with more noise. However, it was also ensured that in the labels no bees were touching, even if that meant making the labeled bees one voxel smaller all around than they appeared to be. These measures were taken with the hopes that the neural networks would 1) learn to connect all three segments of a bee and not rely solely on the grayscale value of the voxel to determine the presence of a bee and 2) learn not to connect bees together. Labeling was completed for one cropped cube measuring 46x46x46 voxels (Fig. 2) and one small swarm (Fig. 3). The cropped cube was completed first and used for the first round of neural network training and development. Though this cube was small, it contained around 50 individual bees and did not have much noise. The small swarm took longer to complete, but once finished was used for training the second round of neural networks. This swarm contained 425 bees, though many were extremely blurry and hard to identify.

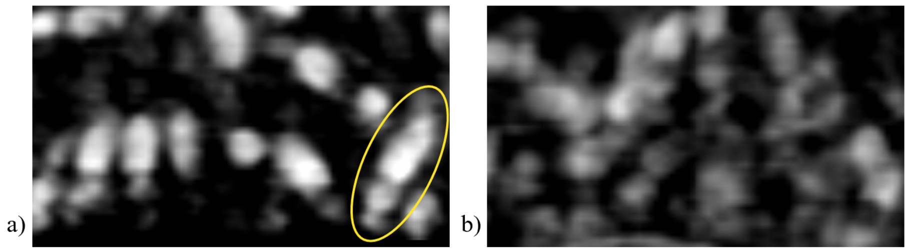

Figure 1: Captured X-ray data from a small swarm of bees. In X-ray images, bees most often show up as three separate segments; an example of a whole bee is circled in yellow where the three segments can be seen. In a single swarm, areas are present with a) well-defined bees and b) high levels of noise, most likely caused by movement of the bees and resulting in blurred bodies. Additionally, bees are often packed tightly together, as shown on the bottom left of image a.

The neural network used for this project has undergone many modifications and will continue to be developed with the hope of improving its performance. Currently, over 40 different networks have been trained, but they all share some basic features: they have a U-Net structure, they take a 32x32x32 voxel X-ray image as an input, and they output a probability map of bee locations, which can then be thresholded to find binary information about the presence of a bee in each voxel (Fig. 4). The first 14 networks, trained on just the labeled cube, provided a benchmark for model performance. The models were primarily evaluated by comparing the binary cross entropy (BCE) between the network output and the provided labels (referred to as loss), comparing residuals, and by visually observing how the model's predictions compared to the target labels. The BCE loss function, as provided by *PyTorch*, can be described as

$$l(x,y) = \{l_1, ..., l_N\}^\top, l_n = -w_n[y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n)],$$

where N is the batch size. This loss function was chosen due to the nature of the data, with binary labels being compared to a probability falling between 0 and 1. Residuals of the network were plotted by subtracting the target labels from the neural network output to compare which voxels were different, an example of which can be seen in figure 6.

Once the small swarm was completely labeled, networks could then be trained with this larger amount of training data. Not only is the small swarm over 60 times the size of the previously used cube, but it also contains a variety of bees, exposing the network to more situations, and consequently improving it's ability to identify previously unseen bees. While this network is not finalized, various ideas have been implemented and the training process seems to have improved over time (Fig. 5). Tested methods of improving the neural network include, but are not limited to, data augmentation, changing the depth of the network, changing the number of channels in each layer, changing the learning rate, adding batch normalization, implementing early stopping with varying patience levels, and adding dropout.

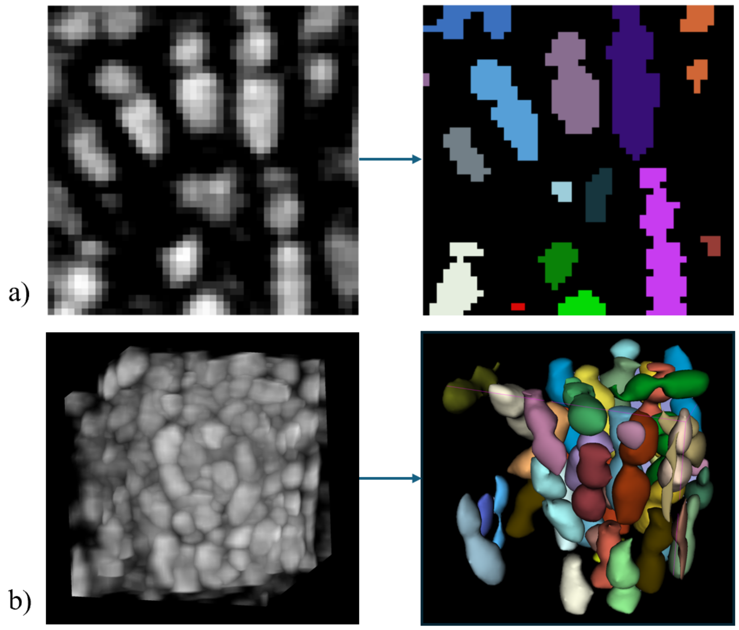

Figure 2: a) A single slice and b) a full three-dimensional cube comparing the original X-ray data to the individually labeled bees. The cube shown in b was the training data for the first round of neural networks developed.

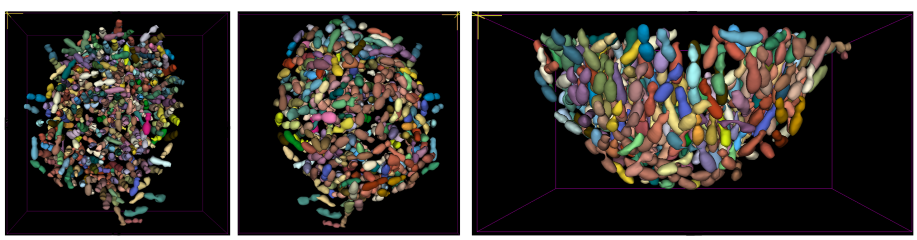

Figure 3: The fully labeled small swarm used for training, visualized in *3D Slicer* from the base (left), the tip (middle), and the side (right). This small swarm contains 425 bees and is approximately ten times smaller than the regular-sized swarms that the network will ideally work on.

Once a neural network has been trained, it can be applied to data that exceeds the 32x32x32 voxel input size in patches. This algorithm takes a certain stride (16 was most often used) and a certain patch size (32 to match the model input) and "walks" along the whole input data, applying the model in patches. Since the data is being cut into smaller, overlapping cubes, it can be safely assumed that, in general, the inside of the patch will contain whole bees while the outskirts will contain bees that have been cropped. Therefore, it is more likely that the network will be able to correctly identify bees in the center of

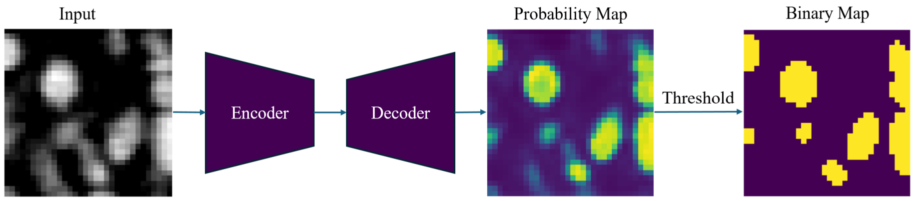

Figure 4: The general architecture of the neural networks. The X-ray input is fed through an encoder-decoder U-Net, which outputs a probability map. This can then be converted into a binary map using a specified threshold.

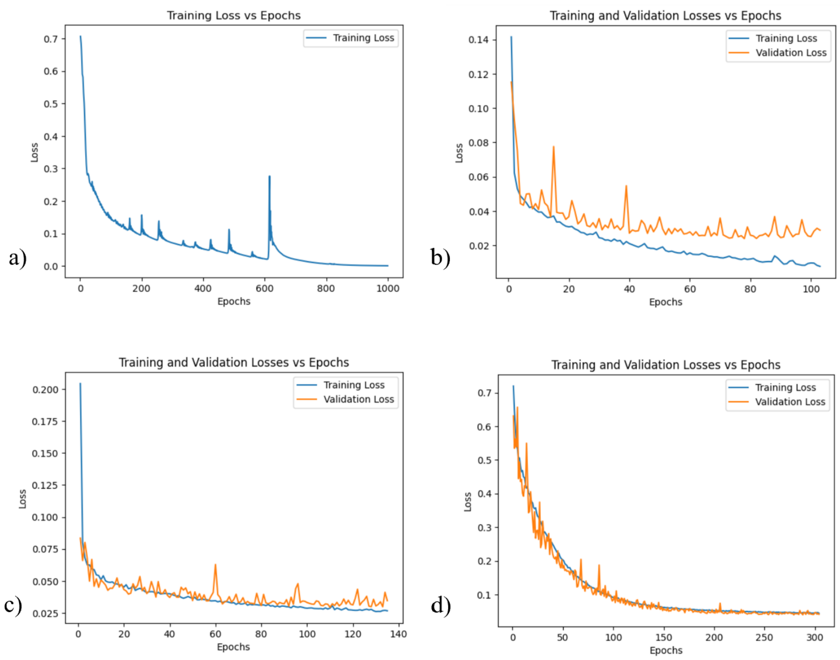

Figure 5: a) The starting networks were trained over 1,000 epochs with all data being dedicated to training. b) To help prevent overfitting, 20% of available training data was set aside for validation data to implement early stopping in later models. c) To further prevent overfitting, dropout was added. d) The learning rate of the models were fine-tuned to prevent a sharp drop-off and quick plateau.

each patch than on the edges, so each patch is weighted with a Gaussian curve, applied in all three dimensions, so the voxels in the center of the patch are weighted more heavily than those along the perimeter. Once all of the patches have been evaluated and normalized, they are combined, and the whole image has been evaluated. Due to the overlapping nature of the patches and the Gaussian normalization, it is believed that the transitions between patches are theoretically seamless.

Once the network has outputted a probability map of where bees are likely to be, a binary map can be obtained after thresholding. Potential threshold values range from 0.1 and 0.9 and dictate how much of the probability map is kept; a threshold of 0.1 will result in dark gray voxels being identified as a bee, while a higher threshold will ensure only very light voxels are kept as bees in the binary map. Varying thresholds dramatically impact the resulting binary map as well as the connected components identified (Fig. 6). Ideally, a threshold would be found which results in each bee being a single component; too small of a threshold will result in different bees being combined while a high threshold might separate one bee into multiple different parts. After finding an

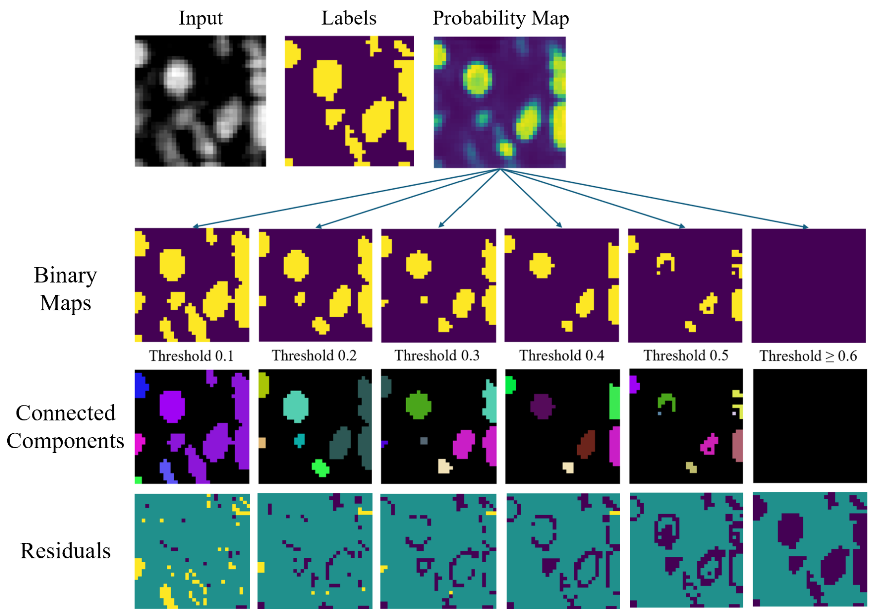

Figure 6: Given a network's output probability map, different thresholds can result in vastly different binary maps and connected components. As seen with a threshold of 0.1, a low threshold can result in separate bees being combined into one connected component (shown in purple). Comparatively, with higher thresholds, a single bee can be divided into separate components, as seen on the right with thresholds 0.3 and higher. Residual plots are also created to compare the network's output to the original labels to help evaluate network performance. Purple pixels indicate areas where bees were found in the labels but not in the output binary map, yellow pixels indicate the opposite, and the blue area is where both the labels and binary map were the same. Given thresholds 0.2 and higher, this network tends to miss voxels around the perimeter of the bees and labels less of the bees as the threshold increases.

appropriate threshold, the location of each bee should theoretically be known, though the network is bound to make some errors. At this point, a plethora of analysis options become possible to better understand the inner working of bee swarms.

4 Results

Out of the 14 preliminary networks trained on just the small cube, the best performing one, given the previously specified criteria, (network 7) had a loss of 0.148 and, when applied to a regular-sized swarm of bees, appeared to segment the majority of the swarm well (Fig. 7). The outer edges of the swarm were, unfortunately, combined into mainly one large component, though this is unsurprising. Images obtained through X-ray Computed Tomography are known to have a higher resolution around the center of rotation, so the bees along the perimeter of the swarm tend to become elongated and blur together slightly due to the nature of how the data is acquired. Additionally, this area tends to be brighter, which the neural network most likely interprets as the presence of more bees. These factors, present in the larger swarm but not in the training data, make it understandable why the network would perform worse in this area. Additionally, the swarm these networks were tested on was a different temperature compared to the training data, so the bees might be orienting themselves in unseen ways. But while this network mostly performed well, it is important to note that the loss might be deceptively low; this network was trained using 27 unique 32x32x32 voxel crops of the labeled 46x46x46 voxel cube and tested on an unseen crop of the same cube. This was the only way to quantitatively measure the loss when only one cube of labeled training data was available for use, but due to the similarity of the training and testing data, it is likely that this model is overfit to this data and would be less capable of generalizing to unseen conditions, such as areas of high noise or bees in varying orientations. This network had the architecture shown in figure 4 but, compared to other networks, had decreased channel sizes.

At the moment of writing, the network trained on the small swarm with the lowest loss (0.246 when tested on the original cube), network 15, appears to perform worse when segmenting a larger swarm, compared to the network described above (Fig. 7). This may be because, as previously mentioned, the labeled small swarm contained areas that appeared to be mostly noise where no distinct bees could be found (Fig. 1). Despite the lack of distinction, individual bees were still identified within the noise and were labeled to try to train the network to be able to pick up on less recognizable bees. Therefore, it is possible that, when given areas of noise, this network is more likely to try to identify bees within the noise when compared to its predecessor. Therefore, when presented with a swarm that has large levels of noise, especially around the perimeters, this network may become overzealous and identify too many bees, creating the one large connected component which can be seen in figure 7. This network's architecture had regular channel sizes, batch normalization, and early stopping with low patience.

The 34th neural network trained has the second lowest loss out of all networks trained on the small swarm. But, despite its marginally higher loss of 0.247, this network appears to be the best at segmenting a regularly sized swarm (Fig. 7). Unlike the other "best" networks identified, this swarm's segmentation does not include any visibly large connected components along the perimeter of the swarm; the majority of bees seem to be separate, but well-formed. Therefore, this network is, out of all of them, the one that most likely provides the best estimation of where each bee in the swarm is. Though improvements can likely still be made, this network's output could already provide interesting insights into the construction of honeybee swarms. This network's architecture had regular channel sizes, batch normalization, dropout, a significantly smaller learning rate, and early stopping with high patience.

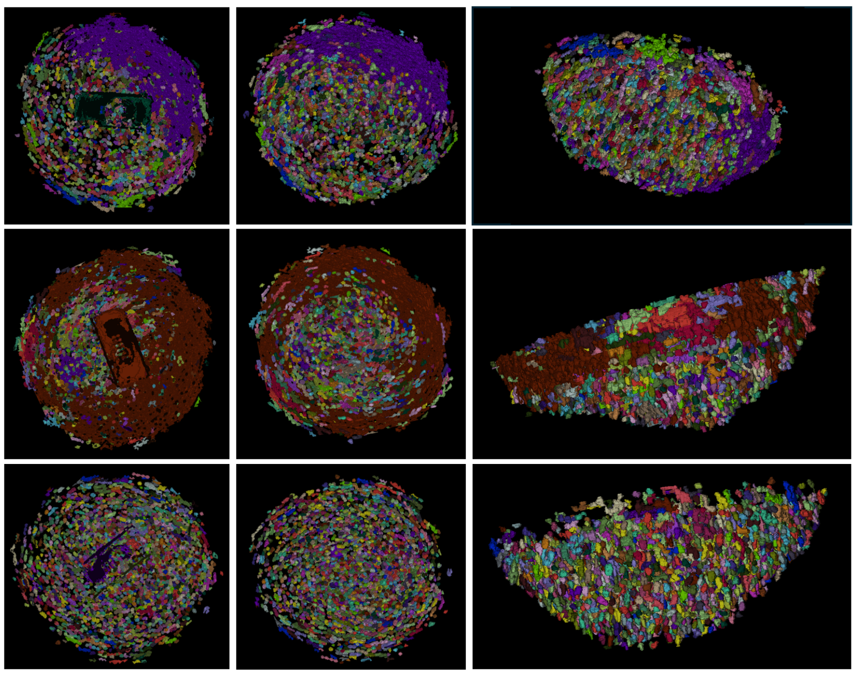

Figure 7: Three-dimensional visualizations of the neural network outputs for a regular-sized swarm with all components smaller than 100 voxels removed. The left shows the base of the swarm with the cage housing the queen bee in the center, the middle is a view pointing up at the tip of the hanging swarm, and the right shows the swarm from more of a side view. a) Network 7 segments the majority of the swarm well, though along the perimeter multiple large components can be seen, especially the large purple region. b) Network 15 also performs worse along the perimeter and base of the swarm, emphasized by the large dark red region. c) Network 34 segments all areas of the swarm well as, unlike the other two networks, there are no large regions of connected bees.

In general, the modifications that most significantly improved the performance of these networks were: adding batch normalization and dropout to each convolutional layer, setting aside validation data and implementing early stopping, and fine-tuning the learning rate. Increased training data is also always beneficial (despite the impressive performance of the network trained on less data), and though, when tested, it did not help these networks, relevant data augmentation should also theoretically improve performance. Despite the overall improved performance of the networks, they all still have their faults; the bees they are trying to segment are irregularly shaped making them hard to learn to identify. Additionally, the disoriented data around the perimeters of the larger swarms caused by X-ray Computed Tomography was unseen in training. Hopefully, going forward, the addition of more training data, particularly focused on the outskirts of a large swarm, will help the networks improve more.

5 Future Work

Though the 34th network appears to segment bees well, more ways of improvement can be explored. Along with various minor adjustments that can be experimented with, it would be interesting to look into using a Mask R-CNN to aid with segmentation, as has been found effective with bacteria segmentation [3]. It would also be interesting to look into the possibility of using an average of outputs from various networks to see if their combined outputs would prove to be better than any one network could produce.

Creating more training data would also likely aid in the performance of these networks. Currently, training a small swarm of bees takes well over 53 hours (assuming 400 bees are in the swarm, labeling each bee takes 8 minutes, the identification of bees is instantaneous, and no quality control is performed), which is quite a time-consuming task. However, training data is vital, so it would be interesting to look into ways to speed up this process. One idea is that, given the current strong performance of some neural networks, the output of these networks can be used as a starting point for labeling. Therefore, instead of starting from scratch and labeling over 400 individual bees, the data can first be fed to the neural network, and the resulting output can be loaded as labels in *3D Slicer*, and then small edits would just have to be made to ensure quality. Theoretically, this could save large amounts of time and dramatically speed up the rate at which the network improves with more training data. Additionally, since many networks have proven themselves to be good at segmenting the centers of swarms but not the perimeters, it would be interesting to see if training data acquired from these outer edges would help with this performance issue. Also, training data from swarms at different temperatures would likely help the networks generalize to swarms in varying conditions.

Lastly, now that there is a way to locate individual honeybees in swarms, even if it is not perfect, this information can be used to help better understand how individual bees contribute to changes in the swarm. It would be interesting to segment swarms at various temperatures, for instance, to compare their makeup. Once the location of each individual bee is obtained, their centers of mass can be found which can be used for various interesting analysis methods as well as informing the distribution of bees throughout the swarm. In the future, maybe this network could even be applied to three-dimensional video footage of the swarms, which would help inform how bees travel throughout the swarm or move to change the swarm's structure. Overall, this segmentation capability could be applied to a variety of research questions in the future and will likely increase our fundamental understanding of honeybee swarms.

References

[1] Adrien Hallou et al. “Deep learning for bioimage analysis in developmental biology”. In: *Development* 148.18 (2021), dev199616.

[2] Bernd Heinrich. “The mechanisms and energetics of honeybee swarm temperature regulation”. In: *Journal of Experimental Biology* 91.1 (1981), pp. 25–55.

[3] Tomasz Konopczyński et al. “Instance segmentation of densely packed cells using a hybrid model of U-net and mask R-CNN”. In: *Artificial Intelligence and Soft Computing: 19th International Conference, ICAISC 2020, Zakopane, Poland, October 12-14, 2020, Proceedings, Part I 19*. Springer. 2020, pp. 626–635.

[4] Orit Peleg et al. “Collective mechanical adaptation of honeybee swarms”. In: *Nature Physics* 14.12 (2018), pp. 1193–1198.

[5] Jacob M Peters, Orit Peleg, and L Mahadevan. “Thermoregulatory morphodynamics of honeybee swarm clusters”. In: *Journal of Experimental Biology* 225.5 (2022), jeb242234.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.

[7] Olga Shishkov et al. “Strength-mass scaling law governs mass distribution inside honey bee swarms”. In: *Scientific Reports* 12.1 (2022), p. 17388.