Proceedings of the 9th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'23)
Brunel University, London, United Kingdom - August 03-05, 2023
Paper No. MVML 120
DOI: 10.11159/mvml23.120

Automatic Detection of Honey in Hive Frames using Deep Learning

Abigail Paradise Vit[1], Yarden Aronson[1]
[1]Department of Information Systems, The Max Stern Emek Yezreel College,
Israel
abigailp@yvc.ac.il

Abstract **-** In recent years, smart technology has become increasingly useful for monitoring honeybee colonies' health and condition in real time using a remote monitoring system. Due to the development of new technologies, it is possible to utilize deep learning techniques in order to improve the understanding of honey conditions within hives. In this study, we propose a method for automatic honey detection in honeycomb frames. A dataset of images of hive frames was collected and annotated by experts. We employed transfer learning by fine-tuning several pre-trained convolutional neural network (CNN) architectures using the image dataset. The best-performing image classification model was VVG19 with an accuracy of 84% and an F1-score of 84% on the test set. As demonstrated in this study, transfer learning can be a useful method of analysing images remotely without human intervention or physical access to remote beehives. Manpower requirements could be reduced and productivity could be improved, particularly in rural areas.

Keywords**:** Image Processing, Deep Learning, Image Classification, Transfer Learning, Honey Production

1. Introduction

Bees play a crucial role in biodiversity since they are crucial for life's survival [1]. As pollinators, bees contribute to ecosystems and food security by producing 35% of agricultural production globally [2].

Beekeepers have raised and maintained honeybees (Apis mellifera) for thousands of years [3]. Beekeeping contributes to sustainable development in the agricultural and food production industries [4-6]. Beekeepers are primarily involved in pollination services and honey production. This includes tasks such as inducting wild swarms, splitting colonies, producing queens, collecting honey, and ensuring the overall health of the hive [7]. According to the European Commission (2020), the EU honey market is valued at $1.4 billion, managing approximately 20 million colonies of honeybees and producing approximately 218,000 tons of honey each year. Due to the importance of the beekeeper industry, it must be provided with additional resources, including providing new utilities and developing mechanisms to ensure beekeeper activities are maintained [8].

Beekeepers typically place their hives in remote fields, and to determine whether the honey is ready for harvest, they physically inspect the remote hive. Beekeepers must identify hives with low honey production over time. This may indicate a colony problem, such as a disease. Furthermore, identifying hives that produce enough honey for harvesting is crucial. This is because a hive already full of honey may not have enough space for the bees to utilize available nectar flows and convert them into honey. The term "honey bound" refers to the condition in which a hive is overflowed with honey and the queen is severely limited in space for egg laying. When a colony reaches this stage, or even before they are fully honey-bound, swarming is more likely to occur. During swarming, a single colony divides into two or more separate colonies. It is the beekeeper's responsibility to recognize this and split the hive into two, add a queen to the new hive, or add honey supers to give the bees more space to grow and produce honey [9,10].

Connected hives have become increasingly useful in recent years with sensors such as humidity and temperature, as well as cameras that can be used to continuously monitor colonies' health and conditions [11-13].

In this study, we propose a method for automatic honey detection in honeycomb frames using images of hives. A dataset of images of hive frames was collected and annotated by experts. We assessed the ability of several pre-trained convolutional neural network (CNN) architectures to detect frames containing honeycombs filled with honey ready for harvest. The performance of the proposed classification model was assessed using accuracy and F1-score metrics. The best-performing image classification model was VGG19 with an accuracy of 84% and an F1-score of 84%. This study shows that transfer learning can be a promising technique for analyzing images remotely and without human intervention or physical access to remote hives. This can reduce manpower needs and improve productivity, particularly in rural areas.

2. Method

Figure 1 illustrates the main phases of our honey detection methodology. Detailed descriptions of the phases will be provided in the following subsections. The process involved the collection of a database of images, and the division of the data into training, validation, and testing sets. A variety of pre-trained convolutional neural network (CNN) architectures were trained and evaluated using training and validation data. Once the most optimal model had been selected, a fine-tuning process was conducted, and the model was tested on unseen data.

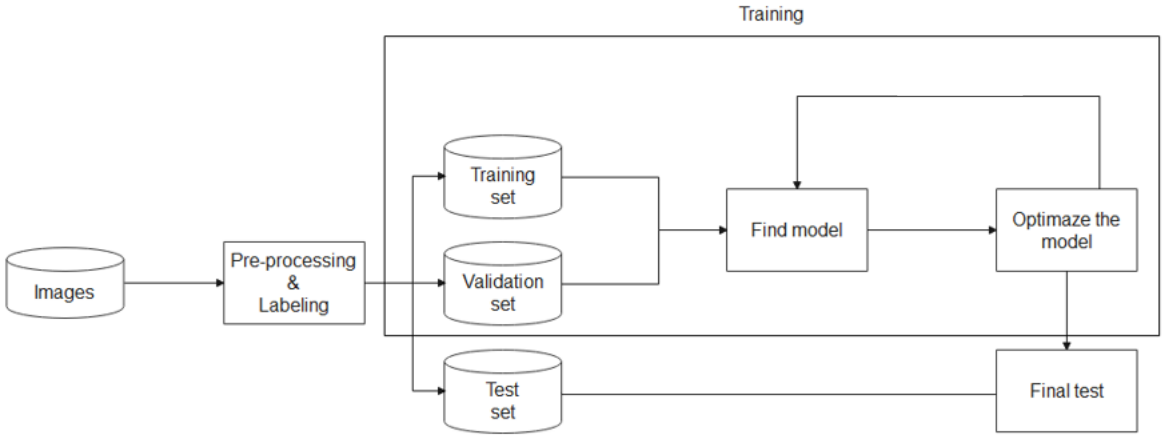

Fig. 1: An overview of the proposed methodology for detecting honey.

2.1. Data

This study collected experimental data from an Israeli honeybee farm (Apis mellifera, the western honeybee). Data was gathered between August 2022 and 6 December 2022. This database contains images taken with a variety of cameras (iPhone 11, Canon 6D, Canon R5). During image preprocessing, each hive was divided into frames. A hive frame is a structural component of a beehive that holds the honeycomb. The number of frames in each hive ranges from six to ten. Figure 2 shows an example of a hive with 10 frames.

Fig. 2: Hive with 10 frames.

A total of 541 frames were captured. Images were taken from an upper hive box (called a honey super) placed over the brood area of a colony. A surplus of honey is stored and harvested in this area [14]. Figure 3 shows a frame with an empty honeycomb before the bees fill it with honey. Figure 4 below is an overhead view of the frame.

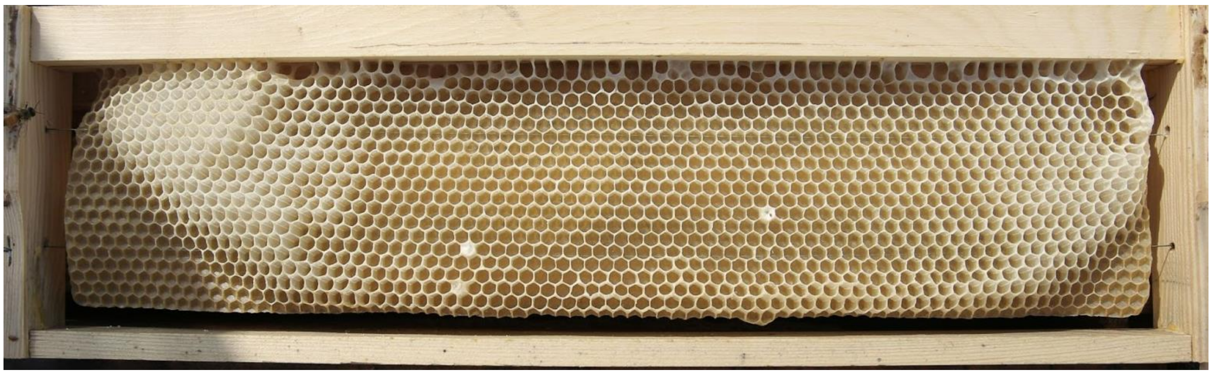

Fig. 3: Empty honeycomb frame.

Fig. 4: An overhead view of an empty honeycomb frame.

A total of 541 frames were captured. Images were taken from an upper hive box (called a honey super) placed over the brood area of a colony. Honey for harvesting is stored in this area [14]. Figure 3 shows a frame with an empty honeycomb before the bees fill it with honey. Figure 4 below is an overhead view of the frame.

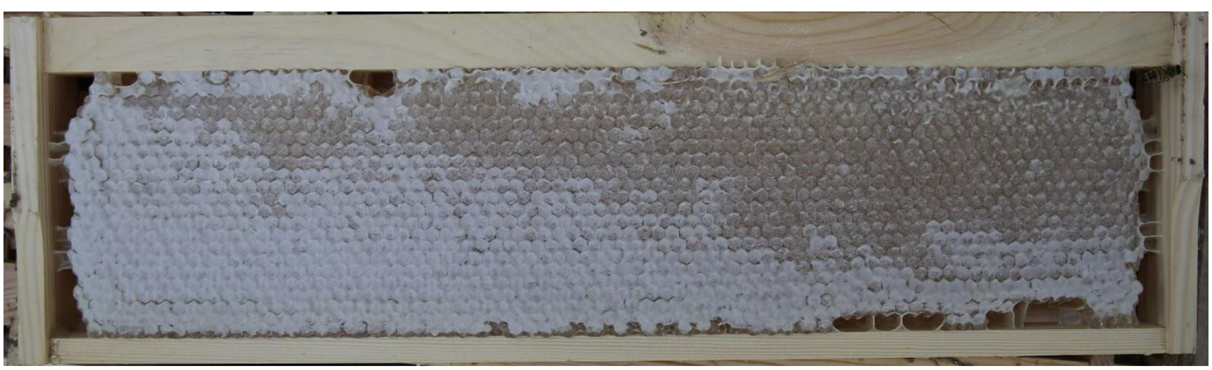

Fig. 5: A honeycomb frame filled with honey.

Figure 6 displays 3 frames from an overhead view, whose honeycombs are full of honey. As can be seen, the honeycombs are visible from both sides of the frame. If the bees run out of space in the honeycombs, they place beeswax in other areas of the hive. This can be seen in comparison to Figure 4, where the honeycomb does not extend out of the frame and is therefore not fully filled.

Consequently, when examining a frame in the hive, it is possible to determine if the honeycomb in the frame is already containing honey (by observing the honeycomb extending out of the frame as well as the beeswax filling the frame).

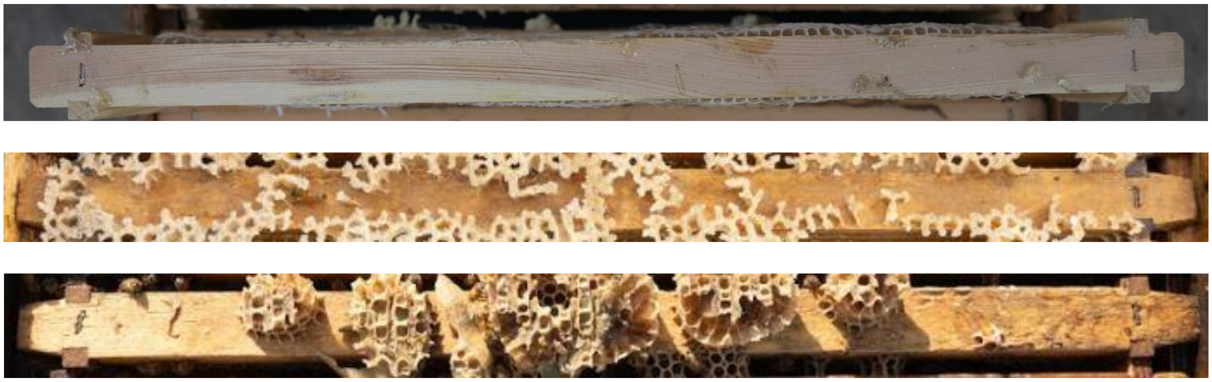

Fig. 6: An overhead view of three frames containing honey.

2.2. Data pre-processing and annotation

Raw image data were preprocessed before being processed by deep learning models. Pre-processing involves three stages: cutting, labeling, and resizing the images.

In the cutting process, each hive image was divided into frames. Since each frame in the hive could receive a different classification, we focused our model on a classification of a specific frame rather than the entire hive.

The next step was to resize each image, with the goal of generating images with a size of 224x224x3. These sizes were necessary for the transfer learning neural network to be constructed.

At the final stage, each image was annotated by a beekeeper expert. The beekeeper usually harvests honey from the frames after the bees have covered at least 75 percent of the honeycomb cells with wax. A low percentage of covered cells may indicate that the honey has not been dehydrated adequately and has not passed through the complete enzyme biochemical process [15].

Each honeycomb frame in our experiment was classified as follows by a beekeeper expert:

0. Less than 75% of honeycomb cells are empty.
1. More than 75% of honeycomb cells are covered.

A total of 181 frames were classified as 0 and 360 frames were classified as 1.

2.2. Convolutional neural network (CNN) architectures

Deep learning is characterized by its ability to create and extract features from raw representations of data. All of this is done without being explicitly instructed on which features to use or how to extract them. Deep learning has been proven to be a powerful technique that can be applied to a wide range of vision-based tasks such as image classification and object detection [16]. Deep learning has been expanded to agricultural fields, including honeybee studies [17]. For instance, deep learning has been applied to estimate the amount of nectar that is produced in images of flowering vegetation using CNN models [18], to identify and count the number of honey bees by conventional networks [19], to classify honey bee comb cells [20], to detect pollen-bearing honey bees in hive entrance images and videos with CNN-based classifiers [21,22] and to detect bee diseases [23-25].

In the context of honey identification in hive frames, we anticipate that utilizing a convolutional neural network (CNN) architecture can automatically extract features from training data that can be used to classify unseen data. Consequently, honey identification in each frame will no longer require beekeeper assistance.

CNN models can be constructed from scratch or from pre-trained models. When there is limited data, training CNNs for computer vision tasks is a common challenge [26]. Compared to building a CNN model from scratch, transfer learning can be a more efficient method since it requires less data and takes less computation time. In our study, transfer learning was preferred due to insufficient images available as inputs. Transfer learning starts with a pre-trained model and fine-tunes its weights using a new dataset. As a result of fine-tuning the pre-trained model on the new dataset, the model learns relevant patterns and relationships relevant to the specific task [26].

A number of CNN architectures have been explored in this study, such as VGG19 [27], DenseNet121 [28], EfficientNetV2S [29], ResNet50 [30], and InceptionV3 [31]. A CNN architecture known as VGGNet was proposed in 2014 [27]. VGGNet is known for its simplicity. The architecture is very uniform, consisting of several convolutional layers with small 3x3 filters, followed by maximum pooling layers and fully connected layers. The network depth is determined by the number of convolutional layers, VGG16 and VGG19 with 16 and 19 weight layers, respectively.

Huang et al introduced DenseNet121 in 2016 [28]. It belongs to the DenseNet family, which stands for "Densely Connected Convolutional Networks". DenseNet models are popular due to their efficient parameter use, improved gradient flow, and state-of-the-art performance. "121" in DenseNet121 refers to the number of layers in the network. DenseNet consists of multiple dense blocks, each consisting of densely connected layers. Each layer is connected to the other in a feed-forward fashion.

EfficientNetV2S, a variant of EfficientNetV2, was proposed by Google researchers in 2021 [29]. With fewer parameters, EfficientNetV2S maintains high accuracy in various computer vision tasks. Compound scaling, stochastic depth, and inverse square root scaling are used in the architecture.

ResNet50 is a CNN architecture proposed by Microsoft Research in 2015 [30]. ResNet50 is a variant of the ResNet architecture, stands for "Residual Network", and addresses the problem of vanishing gradients. ResNet50 architecture consists of 50 layers, including convolutional, pooling, and fully connected layers. It also includes skip connections, which bypass one or more layers and connect the input directly to a later layer.

Inception-v3 was developed by Google Research and consists of several layers of convolutional and pooling operations, as well as auxiliary classifiers at intermediate layers. It uses a technique called "inception modules", which involves parallel convolutions of different sizes followed by concatenation of their output features. As compared to VGGNet, Inception Networks are more computationally efficient [31].

2.3. Experiment Setting

A number of pre-trained CNN architectures have been studied under various configurations to determine the most effective classifier. The experiments were conducted with one Intel Core i7-10510U (16GB RAM).

We divided the images dataset randomly into 90% training images and 10% test images. The training data was re-divided into 80% training and 20% validation. As part of our pre-trained model, the convolutional layer has been frozen. A fully-connected layer has been created, with an output layer consisting of two neurons (one for each class).

Training the model involved 100 iterations. However, we stopped the process when performance on a validation dataset declined to prevent overfitting. Our model was evaluated using validation loss as a stopping criterion. Losses were calculated using cross-entropy.

As part of the first experiment, we examined the models of DenseNet121, InceptionV3, ResNet50, VGG19, and EfficientNetV2S with the default settings of each model (Dropout = 0.2, Number of neurons in middle layer=1024, activation function = relu, activation function in output layer = sigmoid). Since our data was unbalanced, we used weighted measures to ensure the results reflected the frequency of the classes. The model with the highest weighted F1-score on the validation set was selected. To improve the performance of the best model, a fine-tuning process was undertaken. Several configurations were evaluated. We tested the combination of different optimizers, different dropout rates and different learning rates. Additionally, in the fully-connected middle layer, we modified the number of neurons. The model with the highest weighted F1-score on the validation set and the lowest loss score was selected.

In the final step, once the best model has been selected, we test the final model using the test set.

3. Results

We employed various CNN architectures and evaluated their performance in validation set using the following metrics: accuracy, weighted precision, weighted recall, and weighted F1-score. The VGG19 model achieved the highest scores among all other models as shown in Table 1. The VGG19 model was identified as the top-performing model with accuracy of 86% and a weighted average F1-score of 86%.

Table 1: Performance of classifiers

Model	Accuracy	Recall	Precision	F1-score
VGG19	**0.86**	**0.86**	**0.86**	**0.86**
DenseNet121	0.85	0.85	0.86	0.85
EfficientNetV2S	0.84	0.84	0.85	0.84
InceptionV3	0.81	0.81	0.81	0.81
ResNet50	0.79	0.79	0.82	0.77

As VGG19 produced the best results, the model was fine-tuned using several configurations. Table 2 presents the top five models with the highest F1-score in the validation set. Models 1 and 5 had the highest F1 scores, whereas model 5 had a lower loss. Therefore, model 5 was selected as the best model.

Table 2: Fine-tuning classifiers' performance

Model number	Number of neurons	Dropout	Learning rate	Accuracy	Recall	Precision	F1-score	Loss
1	64	0.2	0.005	0.88	0.88	0.88	0.88	0.61
2	64	0.2	0.00001	0.87	0.87	0.87	0.86	0.542
3	64	0.5	0.00001	0.87	0.87	0.87	0.87	0.45
4	128	0.1	0.005	0.87	0.87	0.87	0.87	0.66
5	**128**	**0.5**	**0.00001**	**0.89**	**0.89**	**0.89**	**0.89**	**0.55**

In Figure 7, a learning curve has been provided for the average accuracy and loss during training and validation epochs. As depicted, training and validation accuracy displayed an upward trend with rapid improvement during the initial training stages, while progress slowed down during the later stages. The model's accuracy is increasing steadily, which indicates it is learning and performing well. Overfitting was prevented by terminating the training process when validation accuracy decreased. Both training and testing losses decreased significantly in the early stages. Similarly to the validation accuracy curve, the training process was terminated to prevent overfitting.

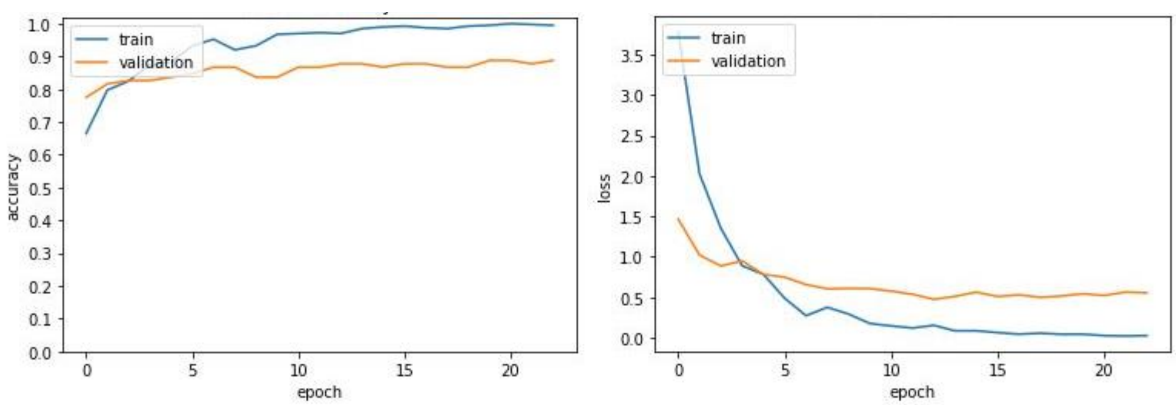

Fig. 7: Training and validation accuracy and loss for best classifier.

For the purpose of evaluating our classifier's ability to generalize unseen images, we saved 10% of the data (55 images) and applied our best VGG19 model to the samples. Model accuracy was 84%, weighted recall was 84%, precision was 85%, and F1-score was 84%.

3. Conclusion

In this paper, we propose a method for automatic honey detection in honeycomb frames. For this task, we collected RGB RGB images of hive frames with expert annotations. The dataset was used for evaluation purposes. Using transfer learning, learning, several pre-trained deep convolutional neural network architectures were compared for fine tuning. VGG19 achieved an overall accuracy of 84% in classifying 55 previously unseen images from the test set. The results indicate that that transfer learning is a promising method for identifying honey in hive frames with high accuracy.

In the future, a greater number of images of hives will be collected and added to the image database.

References

[1] M. L. Winston, *Biology of the honey bee.* Cambridge, Mass: Harvard University Press, 1987.

[2] M. A. Aizen, L. A. Garibaldi, S. A. Cunningham, and A. M. Klein, “How much does agriculture depend on pollinators? Lessons from long-term trends in crop production,” *Annals of Botany*, vol. 103, no. 9, pp. 1579–1588, Apr. 2009.

[3] E. Crane, *The world history of beekeeping and honey hunting*. New York: Routledge, 1999.

[4] W. Ritter and P. Akratanakul, *Honey bee diseases and pests: a practical guide*, 2006.

[5] C. Kremen, N. M. Williams, R. L. Bugg, J. P. Fay, and R. W. Thorp, “The area requirements of an ecosystem service: crop pollination by native bee communities in California,” *Ecology Letters*, vol. 7, no. 11, pp. 1109–1119, Sep. 2004.

[6] S. L. R. Wood, S. K. Jones, J. A. Johnson, K. A. Brauman, R. Chaplin-Kramer, A. Fremier, E. Girvetz, L. J. Gordon, C. V. Kappel, L. Mandle, M. Mulligan, P. O'Farrell, W. K. Smith, L. Willemen, W. Zhang, and F. A. DeClerck, “Distilling the role of ecosystem services in the Sustainable Development Goals,” *Ecosystem Services*, vol. 29, pp. 70–82, Feb. 2018.

[7] A. Mizrahi and Y. Lensky, *Bee Products: Properties, Applications, And Apitherapy.* Springer Science & Business Media, 2013.

[8] O. Etxegarai-Legarreta and V. Sanchez-Famoso, “The Role of Beekeeping in the Generation of Goods and Services: The Interrelation between Environmental, Socioeconomic, and Sociocultural Utilities,” *Agriculture*, vol. 12, no. 4, p. 551, Apr. 2022.

[9] M. Bencsik, J. Bencsik, M. Baxter, A. Lucian, J. Romieu, and M. Millet, “Identification of the honey bee swarming process by analysing the time course of hive vibrations,” *Computers and Electronics in Agriculture*, vol. 76, no. 1, pp. 44–50, Mar. 2011.

[10] J. C. Loftus, M. L. Smith, and T. D. Seeley, “How Honey Bee Colonies Survive in the Wild: Testing the Importance of Small Nests and Frequent Swarming,” *PLOS ONE*, vol. 11, no. 3, p. e0150362, Mar. 2016.

[11] C. Chen, E. C. Yang, J. A. Jiang, and T. T. Lin, “An imaging system for monitoring the in-and-out activity of honey bees,” *Computers and Electronics in Agriculture*, vol. 89, pp. 100–109, Nov. 2012.

[12] F. Edwards-Murphy, M. Magno, P. M. Whelan, J. O’Halloran, and E. M. Popovici, “b+WSN: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring,” *Computers and Electronics in Agriculture*, vol. 124, pp. 211–219, Jun. 2016.

[13] P. Marchal, A. Buatois, S. Kraus, S. Klein, T. Gomez-Moracho, and M. Lihoreau, “Automated monitoring of bee behaviour using connected hives: Towards a computational apidology,” *Apidologie*, vol. 51, no. 3, pp. 356–368, Dec. 2019.

[14] G. O. Babarinde, S. A. Babarinde, D. O. Adegbola, and S. I. Ajayeoba, “Effects of harvesting methods on physicochemical and microbial qualities of honey,” *Journal of Food Science and Technology*, vol. 48, no. 5, pp. 628–634, Mar. 2011.

[15] R. Goodman and P. Kaczynski, *Australian beekeeping guide*. Barton, A.C.T.: Rural Industries Research and Development Corporation, 2014.

[16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[17] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, Apr. 2018.

[18] D. Hicks, M. Baude, C. Kratz, P. Ouvrard, and G. Stone, “Deep learning object detection to estimate the nectar sugar mass of flowering vegetation,” *Ecological Solutions and Evidence*, vol. 2, no. 3, Jul. 2021.

[19] V. Kulyukin and S. Mukherjee, “On Video Analysis of Omnidirectional Bee Traffic: Counting Bee Motions with Motion Detection and Image Classification,” *Applied Sciences*, vol. 9, no. 18, p. 3743, Sep. 2019.
[20] T. S. Alves, M. Alice Pinto, P. Ventura, C. J. Neves, D. G. Biron, A. C. Junior, P. L. De Paula Filho, and P. J. Rodrigues, “Automatic detection and classification of honey bee comb cells using deep learning,” *Computers and Electronics in Agriculture*, vol. 170, p. 105244, Mar. 2020.
[21] Z. Babic, R. Pilipovic, V. Risojevic, and G. Mirjanic, “Pollen bearing honey bee detection in hive entrance video recorded by remote embedded system for pollination monitoring,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III–7, pp. 51–57, Jun. 2016.
[22] I. F. Rodriguez, R. Megret, E. Acuña, J. L. Agosto-Rivera, and T. Giray, “Recognition of Pollen-Bearing Bees from Video Using Convolutional Neural Network,” *Workshop on Applications of Computer Vision*, Mar. 2018.
[23] K. Bjerge, C. E. Frigaard, P. H. Mikkelsen, T. H. Nielsen, M. Misbih, and P. Kryger, “A computer vision system to monitor the infestation level of Varroa destructor in a honeybee colony,” *Computers and Electronics in Agriculture*, vol. 164, p. 104898, Sep. 2019.
[24] P. Davidson, M. Steininger, F. Lautenschlager, K. Kobs, A. Krause, and A. Hotho, “Anomaly Detection in Beehives using Deep Recurrent Autoencoders,” *International Conference on Sensor Networks*, Jan. 2020.
[25] C. Liu and S. Lin, “A Pest Intrusion Detection in Chinese Beehive Culture Using Deep Learning,” *Scientific Programming*, vol. 2022, pp. 1–10, Apr. 2022.
[26] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2016.
[27] K . Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014).
[28] G. Huang, Z. Liu, L. Van Der Maaten, & K. Q. Weinberger, (2017). Densely connected convolutional networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).
[29] M. Tan and Q. Le, “Efficientnetv2: Smaller models and faster training,” *In International conference on machine learning* , Jul. 2021.
[30] S. Targ, D. Almeida, K. Lyman. Resnet in resnet: Generalizing residual architectures. arXiv preprint arXiv:1603.08029 (2016).
[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826). 2016.