signals

MDPI

Article

Deep Learning Beehive Monitoring System for Early Detection of the Varroa Mite

George Voudiotis, Anna Moraiti and Sotirios Kontogiannis *

Laboratory Team of Distributed Microcomputer System, Department of Mathematics, University of Ioannina, University Campus, 45110 Ioannina, Greece; ma10635@uoi.gr (G.V.); ma12007@uoi.gr (A.M.)
* Correspondence: skontog@uoi.gr; Tel.: +30-26510-0-8252

Abstract: One of the most critical causes of colony collapse disorder in beekeeping is caused by the Varroa mite. This paper presents an embedded camera module supported by a deep learning algorithm for the process of early detecting of Varroa infestations. This is achieved using a deep learning algorithm that tries to identify bees inside the brood frames carrying the mite in real-time. The end-node device camera module is placed inside the brood box. It is equipped with offline detection in remote areas of limited network coverage or online imagery data transmission and mite detection over the cloud. The proposed deep learning algorithm uses a deep learning network for bee object detection and an image processing step to identify the mite on the previously detected objects. Finally, the authors present their proof of concept experimentation of their approach that can offer a total bee and varroa detection accuracy of close to 70%. The authors present in detail and discuss their experimental results.

Keywords: beehive monitoring systems; Varroa mite detection; Internet of Things (IoT); Convolutional Neural Networks (CNNs); image processing; performance evaluation; distributed systems

check for updates

Citation: Voudiotis, G.; Moraiti, A.; Kontogiannis, S. Deep Learning Beehive Monitoring System for Early Detection of the Varroa Mite. *Signals* **2022**, *3*, 506–523. https://doi.org/10.3390/signals3030030

Academic Editors: Hua-Liang Wei and Zhao-Hua Liu

Received: 15 May 2022
Accepted: 14 July 2022
Published: 28 July 2022

1. Introduction

Bees' population declination worldwide is mainly caused due to the reckless use of pesticides by humans [1], as well as the emergence of new resistant strains of bacteria and mites. Varroa disease is one of the most resistant diseases with two main forms (Varroa destructor and Varroa jacobsoni) [2], affecting the growth of the offspring, stressing the bees, and reducing their chances of survival in winter. More specifically, Varroa mite is one of the most devastating diseases for the beekeeping population. This parasite attaches to the bee and feeds on its fat cells, causing wing deformations.

The environment inside and around the beehives is vital to the colony establishment's success and development. An essential factor in apiary hives that affects both colony survival and honey yield is the ability to manage agricultural interventions and disease treatments (especially Varroa mite [3]) and monitor the conditions inside the beehive [4–6].

Most beekeepers use synthetic pesticides to control the mite infestation; nevertheless, the mite soon develops resistance to their active compounds, compromising their effectiveness. The prevalence of the mite over the year is confronted by: (a) The reproduction of resistant Apis mellifera breeds [7] and (b) the appliance of precise interventions upon the detection of the mite that includes controlled high temperatures (above 38 °C) or increased relative humidity stress events above 70% for normal brood box temperature environments (32–37 °C) [8,9].

The Internet of Everything (IoE) industry is shifting fast towards the beekeeping sector, aiming for the vast applicability of Internet technologies and IoT, capable of applying new smart AI detection algorithms [10–12]. Existing applications in agriculture include environmental monitoring and bee monitoring IoT systems [13–18]. This paper focuses on technological solutions for detecting bees and then Varroa mites using cameras inside the brood box.

Significant improvements have been made toward monitoring beehive conditions using cameras over the last few years. Such systems try to replace systems that use IR sensors at the beehive entrance or sound sensors [13,19–21] and translate sensory information to conditions status. Therefore, image processing techniques and deep learning techniques have been utilized to monitor bees, and their actions at the hive entrances, as mentioned at [22,23]. Image processing methods are the easiest to implement. However, deep learning methods produce the most reliable condition outcome results, such as swarming or external attacks. Nevertheless, if such probing methods are performed only at the beehive entrance, they cannot reliably detect the occurring event early enough for the beekeepers to ameliorate them successfully.

This paper investigates existing technological systems focusing on implementing an IoT system for detecting Varroa mite in bees, which leads to major Colony Collapse Disorders (CCD) around the globe. Since Varroa infestation events occur mainly in the late spring (Apr-May) and first Autumn months, as indicated by the apiarists, this paper presents a new camera sensor system for detecting Varroa inside the beehive. The proposed system uses a camera that incorporates an image processing motion logic and utilizes a Convolutional Neural Network CNN pre-trained model for detecting bees. The rest of this paper is structured as follows: Section 2 presents related work in existing technological products for Varroa mite identification and confrontation. Section 3 presents the authors' proposed Varroa detection system and system capabilities. Section 4 presents the proposed CNN algorithm used by the system. Section 5 presents the authors' system experimentation (offline and online), evaluation, and cross-comparison with existing literature systems. Finally, Section 6 concludes the paper.

2. Related Work

This section presents the related work on detecting and treating Varroa. It is divided into two subsections: The first subsection includes Periodic uniform interventions (treatments) that mainly incorporate chemical substances, and the second subsection describes in detail technological detection systems that try to offer precise mite detection for beekeeping interventions to apply.

2.1. Related Work on Periodic Varroa Mite Treatments

Beekeepers have been evaluating the efficacy of an acaricide treatment by counting dead mites that drop from brood frames and bees onto the hive bottom board. Methods that they use are the "sugar shake" procedure, washing kits (alcohol wash), and organic substances such as thymol [24,25].

Another way of Varroa treatment is by using generated smoke-vaporizer kits produced by acid mixtures (usually Oxalic acid and glycerin). This smoke can be an effective and comparatively cheap method [26]. The use of antigenic acid is also mentioned by [27] as a means of limiting Varroa infestations if sprayed inside the brood box.

Various chemical substances and materials are used to determine if a colony is infested with Varroa mites, collecting and killing the mites. These normally are the following: Ether (or Alcohol) roll, Powdered-sugar shake, chemical sticky-boards (Mite Census), Drone/Brood Sampling, and Sticky-Board with acaricides [27,28].

However, because of the number of negative points in the usage of the chemical methods, the search for more efficient alternative methods kept going with the inclusion of the developing technology.

2.2. Related Work on Varroa Mite Detection Technologies

A technological method reported by [29] uses the E-nose technology. Its use is to detect Varroa, when the infestation influences the chemical composition of the air inside a hive. The deciding factor for the effectiveness of this technique is the time of detection.

Sound monitoring systems using MEL or FFT spectrograms are also powerful tools for Varroa mite indications. The use of frequency-amplitude over time representations combined with SVMs and neural network classifiers can distinguish between different hive states, specifically external attacks, colony stress, swarming, and queen loss [19]. Detection of colony collapse can be exploited as an indicator of the existence of the Varroa mite.

The authors at [13,30,31] also present their implementations of low power Wireless Sensor Network (WSN) technology, assisted by cloud computing, to monitor the existence of bees stress behavior, that also includes the existence of the Varroa mite. The key objective of this research is to use WSN technology to notice a beehive colony and collect key information about the activity/environment within a beehive and the health of the bees. Nevertheless, as mentioned by [13], Varroa mite detection using sensors (temperature, humidity, noise level, and gas sensors), may include many false detection cases, since other phenomena such as swarming, queen loss, or even hunger may lead to CCDs as the ones caused by the mite.

Based on its eco-friendly and sustainable nature, accurate results, and futuristic design, Var-Gor [32] is a promising device for the early detection of Varroa mite as well as its early struggle. Specifically, when a contaminated bee with varroa mite enters an uncontaminated hive, the Var-Gor system detects the mite by image capturing, template matching, color classification, and segmentation filters. Moreover, this sends a warning by notification to the beekeeper’s phone.

For real-time bee monitoring using cameras, including deep learning approaches, the authors at [33] developed an experimental system. Their proposition is built on a single-board computer Raspberry Pi (RPi) platform and aims to analyze video streams with bees and detect varroosis. They also applied two distinct detection processes with two CNN models, one for the bees and the other for the Varroa. Nevertheless, the camera is located outside the hive, making it hard for mite early detection. In case of infection, the pictures of the infected bees are transferred to the data center in the Cloud for further analysis, storing, upgrading their CNN models, and sending a notification to the relevant beekeeper. The use of two distinct CNN models makes this approach hard to apply detection to a standalone RPi device inside the device, as the authors of this paper propose with their offline system approach. Furthermore, the use of image processing techniques such as edge detection Hough transformations, region labeling, and color masking can accurately identify the mite on the detected bees, provided by a CNN trained network that accurately detects bees, similar to the one the authors of this paper propose.

The authors at [34] present a camera-based approach of CNN-trained networks using the classification of infected and non-infected bees manually and then utilizing a laser to exterminate the infected ones. This approach, as presented, has the drawback of using single bee images labeled and classified. This can perform well on detecting single bees on the beehive door openings or white background but fail significantly on detecting bees inside the frames (where the mites reside) due to the vast concentrations of bees on each frame. The solution to this problem proposed by this paper is to use frame images where manual bee annotation and image localization are performed before CNN training. Since regions in the image do not contain any information, image segmentation and manual identification of the ROIs can offer significantly better results.

The authors at [35] aim to provide a solution to identify Varroa mite from low quality and a limited number of images. The proposed model combines an image enhancement method CLAHE, data augmentation method DCGAN, and an optimized classification method CNN to classify infected or healthy bees from standard bee images. The results convey that the CLAHE method improves sharpness and positively affects the CNN performance. Furthermore, the DCGAN augmentation method provided more promising results than the conventional ones in the infection identification scenario. In conclusion, this vision-based approach appears to be more suitable and efficient for identifying Varroa mites on bees.

Finally, the authors at [36] experimented on whether the state-of-the-art object detectors using ill bees and varroa mites annotated datasets. Then, they experimented with CNN algorithms such as YOLOv5 [37] and SSD [38], to perform the varroa mite and ill bee detection. The authors CNN testing using F1-score results have shown a score of 87% for Yolo ill bees detection and above 70% for SSD varroa mite detection. The authors tried to use the Deep SVDD anomaly detector. However, the SVDD anomaly detector was not able to model the problem. The authors also mention that Jetson Nano can be used as part of an implemented detection end-node device. The authors' proposition of using YOLOv5 provides significant results. Nevertheless, no detection performance results are given out. Furthermore, the use of the F1 score metric that represents the model's precision and sensitivity is not an expected value of the model's accuracy. The mAP score indicates a good accuracy model but not a credible accuracy metric.

The following section presents the authors' proposed Varroa Detection system implementing their proposed smart detection algorithm, which focuses on the early detection of the mite to apply a precise treatment.

3. Proposed Varroa Detection System Implementation

The authors propose a new incident response system for the automatic detection of Varroa mite. The system includes the following parts: (1) The end-node device, (2) the cloud service for the online detection process, (3) the concentrator device serving the online detection process, and the mobile phone application used for the offline detection process. The end-node detection device parts and capabilities are presented in detail in the following subsections.

3.1. Beehive Camera End-Node

The beehive camera monitoring module is attached to a plastic beehive frame, and it is placed inside the beehive brood box, as shown in Figure 1. The module includes the following components:

- The camera module component. There are two camera module components included in the end-node detection module. The first is a 5 MP camera with a fisheye lens of 160° sighting and manually adjusted focus. This camera is connected to the end node microprocessor using a 15 pin FFC cable. The first camera is located in the middle of the plastic frame (see Figure 1, Camera 1). There is also a second camera attached to the end-node device. This is a 5 MP USB camera located on top of the plastic frame. It is placed on top of a plastic frame covered entirely with a smooth plastic surface to avoid being built or waxed by bees. It is equipped with a small led and a hinged arm that allows the camera to take images above and on top of the frames. Both cameras are connected to a quad-core ARM microprocessor and can be used concurrently to capture bee images inside the brood box.
- The Microprocessor Control Unit (MCU). The MCU is a quad-core embedded 64 bit-ARM microprocessor device operating at 1 GHz, including a 512 MB LDDR2 RAM clocked at 450 MHz. The MCU is responsible for storing camera snapshots to its embedded SD-card and if appropriately configured, uploading them to the cloud, using the Varroa detection service Application Interface (API) is created for that purpose (see Figure 1, ARM end-node CPU).
- The data transmission modules. There are MCU embedded Wi-Fi and Bluetooth 4.2 with Bluetooth Low Energy (BLE) capable transponders attached to the MCU. The transponders are used in turn by the two modes of end-node device operations: Online and offline.
- The autonomous device power component. It includes a 20 W/12 V PV panel connected directly to a 12 V-60 Ah lead-acid SLA/AGM battery (see Figure 1). The battery is placed under the PV panel on top of the beehive and feeds the ARM MCU unit using a 12 V voltage regulator with 2 × USB outputs, used to power the end-node device through its micro-USB power port. The battery used is a deep depletion one, since the system might get fully discharged due to its short battery capacity, especially at night or on prolonged cloudy days.

- The real-time clock module. An Inter-Integrated Circuit (I2C) DS1307 Real Time Clock (RTC) module is placed at the General Purpose Input Output (GPIO) pin of the MCU for the process of keeping track of time if the end-node device is operating in offline mode. Suppose online appropriate Network Time Protocol (NTP) service is automatically used to calibrate time offsets and reset the RTC DateTime.
- The Wi-Fi concentrator. It is used only in the online mode of operation, and it is a Wi-Fi access point device that includes an LTE/3G cellular transceiver. The end-node MCU connects to the concentrator for the process of imagery data uploads to the cloud if the device operates in online mode. For the offline mode, the MCU BLE interface is used, transmitting to a distance up to 4–10 m the detection output of the beehive. The following subsection describes the end-node device's two modes of operation.

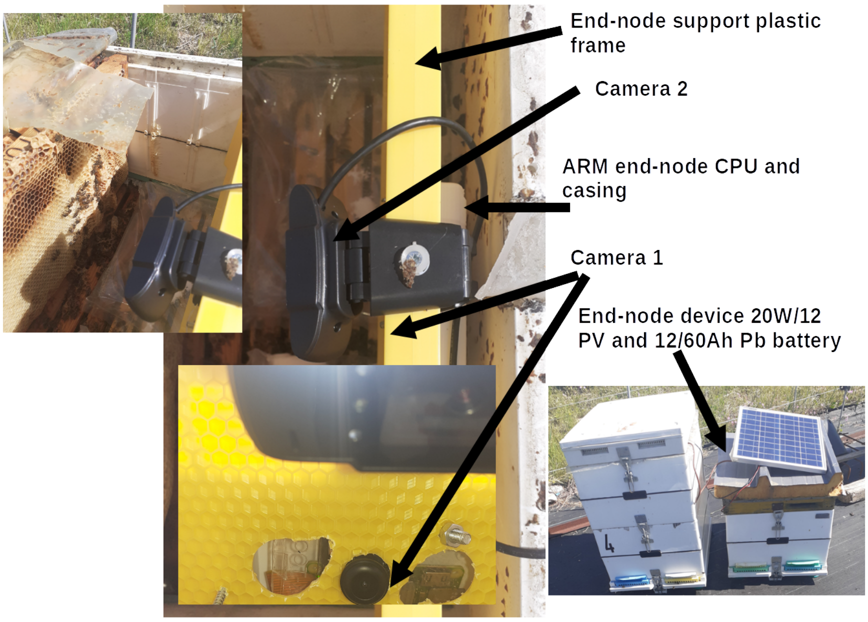
End-node support plastic frame
Camera 2
ARM end-node CPU and casing
Camera 1
End-node device 20W/12 PV and 12/60Ah Pb battery

Figure 1. Proposed Varroa mite detection system end-node device and device components.

3.2. End-Node Device Functionality and Modes of Operation

There are two modes of operation that the end-node device can be used: Offline and online. The operation modes can be selected using a selection switch placed at the bottom of the MCU casing. The switch is connected to an MCU GPIO pin digital input, and by setting it to LOW or HIGH, it switches between the two modes of operation accordingly. The two end-node modes operate as follows:

- **Online mode:** In this mode of operation, the process of Varroa detection is performed over the cloud. For this purpose, appropriate cloud service and API using HTTP requests have been implemented. The API is capable of image data transmission from the uploaded by the end-node MCU using HTTP protocol PUT requests. Also, an HTTP JSON POST request can be sent to the cloud API, including an API key and a beehive id, and the API returns as part of a JSON object. The Varroa detection results for this beehive, including the base64 encoded images Regions Of Interest (ROIs), where Varroa mite has been detected.
  The online mode of operation requires using the beehive concentrator, which is responsible for the node data transmissions over the internet over HTTP. It acts as an intermediate gateway among the end nodes and the cloud application service. The concentrator can upload images with an overall bandwidth capability that varies from 1–7/10–57 Mbps, depending on the gateway distance from the beehive [39]. If the distance is 20–30 m, it is limited by the LTE technology used.

- **Offline mode:** In places of limited Internet connectivity and cellular coverage, the end-node offline mode can be used. The offline mode does not require the use of the concentrator device.
  In this mode, a micro-service that includes a version of the detection algorithm inside the end-node device is used for the process of executing the Varroa detection algorithm locally (as presented in Figure 2). Then, the final CSV output is transmitted using the MCU BLE transponder to the farmers' mobile phones. A BLE service and two read characteristics can be used for the CSV output, and Varroa detected ROI image acquisition accordingly. The beekeeper can check the status of each one of his beehives by moving close to the beehive and pairing with each one accordingly, performing the BLE read from his mobile phone. The drawbacks of the offline mode are that it offers 20–25% less end-node energy consumption and no communication provider costs. Nevertheless, it has difficulties with BLE pairing, especially if many BLE devices are close-by and there are difficulties on characteristic reads of imagery base64 encoded data [40,41]. For this reason, only one Varroa mite ROI is available (the last one detected) via that BLE characteristic.

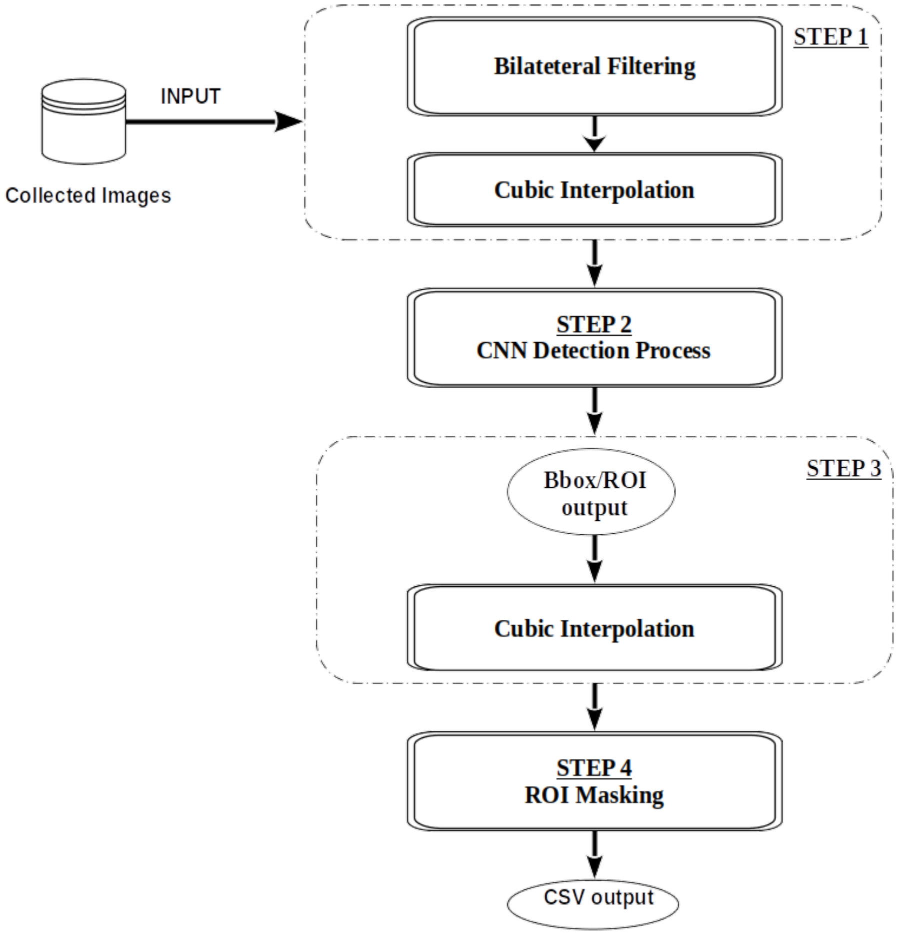

Figure 2. Proposed algorithm process flowchart for the detection of Varroa mite.

The following section describes the Varroa mite detection algorithm used by the end-node devices in offline or online operation modes.

4. Proposed Method for Varroa Mite Early Detection

Training a Convolutional Neural Network for an object initiates with the collection of the sample of images that contain the objects to be detected. If no object localization applies to the image, then the process involved is a classification process that assists in classifying the entire image contents. On the other hand, suppose the localization of multiple objects in the image is enforced. In that case, the constructed process is a detection one, and it may provide output bounding boxes or masked areas of the detected objects. The first is called object segmentation, while the latter is called instance segmentation.

For CNN classifiers or detectors, different models can apply. In this paper, the Regional CNN algorithm Faster R-CNN [42] is used on pre-trained models [43], for the collection of fixed size bounding boxes and confidence scores in the presence of object instances. In these pre-trained models, arbitrary classes can be set, attached, and trained as part of the model's classes. The most well-known algorithms are the Single Shot MultiBox Detector (SSD) [38] and the Faster R-CNN [44]. These algorithms and these used models are the most common because they have been created to balance efficiency and accuracy. The SSD algorithm uses a Convolutional Neural Network (CNN) to input images only once and outputs a feature map [38]. The feature map then goes through a convolutional kernel to predict the bounding boxes and the possibility of classifying them. The Faster-RCNN algorithm uses a small convolutional network called the Region Proposal Network (RPN) to create areas of interest in which the network predicts the probability of being the background or object of interest [44].

For the implementation of their bee-detection process, the authors have selected MobileNet pre-trained CNN models since they are proven lightweight models with short detection times and can be used in embedded systems as part of deep industrial learning real-time or close to real-time applications [45]. Additionally, the authors selected a heavy model such as ResNet-50, with five times more trainable parameters that can be loaded to an embedded device to compare the accuracy results with the ones derived from lightweight models.

In this section, the authors describe their process of detecting the Varroa mite inside the beehive. Their proposition includes two steps: (a) use of SSD and Faster-RCNN Convolutional Neural Networks (CNN) including pre-trained models [46,47] for the detection of bees inside the brood box frames and (b) use of color masking and Hough transformation for the detection of Varroa upon the previously detected bee objects.

The algorithmic process used to build the neural networks and carry out the detection of bees is comprised of five steps. The first four are necessary for the CNN bees' object detection process, and the fifth step is required for the edge detection of the Varroa mite on the previously detected contours. The detection process steps are shown in Figure 2. The proposed algorithm CNN training steps and detection step used are as follows:

- **Step 1—Initial data acquisition and data cleansing:** The initial imagery dataset acquired by the Beehive monitoring module is manually analyzed and filtered to eliminate blur images or images of low resolution and light intensity. The photos in this experimentation taken from the camera module are set of the minimum acquisition of 5 Mpx size of 800 × 600 px 300 dpi compressed at JPEG format using a compression ratio Q = 70, of picture size 350–500 KB each. Similarly, the trained CNN network and algorithms used are the most processing light for portable devices, using a minimum trained image size input of 640 × 640 px (lightly distorted at the image height) and Cubic interpolation.

  The trained Convolutional Neural Network (CNN) is used to solve the problem of swarming by counting bees' concentration above the bee frames and inside the beehive lid. The detection categories that the authors' classifier has used are:

  Class 0: No bees detected.
  Class 1: very small number of bee detection (less than 10).
  Class 2: Small number of bee detection (20–30).
  Class 3: Medium number of bee detection (30–50).
  Class 4: High number of bee detection (more than 50).

  For each class, a number of detected bees has been set as a class identifier (The class identifier boundaries can be arbitrary and set accordingly at the detection service configuration file). Therefore, the selected initial dataset must consist of at least 1000 images per detection class, a total of 5000 images used for our training CNN case.

- **Step 2—Images transformation and manual data annotation:** All images that went through the clearing process were manually annotated using the LabelImg [48] tool. There are other tools used for the photo annotation process, such as Labelbox [49], ImgAnnotation [50] and the Computer Vision Annotation Tool [51], which always create an output in either the JSON or XML format.

  The resolution and clarity of the original images are extremely important, as this facilitates the detection of the Varroa mite. Regarding the clarity of the photo, the method used is as follows. A bilateral filter smothers all images using a degree of smoothing sigma = 0.5–0.8 and a small 7 × 7 kernel. Afterward, all photos must be scaled to particular and fixed dimensions to be inserted into the training network. Scaling is performed either using a cubic interpolation process or a super-resolution EDSR process [52].

  The preparation of the photos is initially based on the dimensions that each training algorithm requires for its smooth operation. The OpenCV [53] library is used for the image transformation process and is part of the second and fifth stages of detection. The second stage is before the detection of bees using CNN, and the fifth is the stage of detection of the Varroa mite (see Figure 2).
- **Step 2—Training process:** The training process is based on the use of PyTorch pre-trained Convolutional Neural Network (CNN) models [54] and the use of all available system resources. The essential computer subsystem for the training process is the GPU to speed up the neural network training. CUDA tools and libraries are used for this purpose according to PyTorch requirements.

  CNN's creation is based on pre-existing PyTorch [54] trained models used to train the neural network to detect bees. The selected PyTorch models and their capabilities for the detection process are presented in Table 1. After the annotation of the images of Step 2 is completed, the images are divided into two sets. The first set is the training set which contains 70–80% of the annotated photos, and the remaining 20–30% is the test set. The second can be divided into 50% to create another set which will be the validations sample. Then, you choose the model that will be used for the training. The output of the training process is the CNN model used in Figure 2, which is the Step 2 detection process of bee objects.

Table 1. Pre-trained CNN models that were used by the proposed faster R-CNN algorithm training process.

CNN Mode	**CNN Models Capabilities**		
	Input Images Size (wxh) [px]	**Accuracy for Detected ROIs Confidence Level Values = 100%**	**Accuracy for Detected ROIs Confidence Level Values ≥ 50%**
MobileNet V2	640 × 640	71.878	90.286
MobileNet V3	640 × 640	74.042	91.340
ResNet-50 FPN	640 × 640	76.130	92.862

- **Step 3—Detected bee contours:** This step includes a selection process of bee-detected objects based on the confidence threshold value set by the service. A good confidence value threshold that can be used is above 0.5 (50%). Then, the Gaussian filtering cubic interpolation is applied to the selected contours to scale them to sizes (wxh) of 40 × 50 px for step 4 to apply on each distinct ROI.
- **Step 4—ROI masking, Varroa mite detection step:** This step includes a color transformation from RGB to HSV. Then, an appropriate HSV mask is applied to each bee scaled image, transforming them into binary images, where the detected by the mask areas are set to white and all other areas to black. Then, a Hough transformation is

applied to detect circular areas with a lower-upper threshold of 10–90 px^2. If at least one circular area of this threshold is detected on a bee object, this bee is set as detected with Varroosis. The original color image, including the detected bee object contours with the masked Varroa mite areas annotated on each CNN detected image, is stored in the appropriate output folder. The detection results are appended to the detection service process CSV output file.

- **Detection service process and data output:** The detection process is performed by a daemon application that is installed as a service on a cloud server or at the embedded end-node device depending on the mode of operation (online, offline). This application loads the inference graph of the CNN neural network into the system memory so that the bees can be detected and then the Varroa mite can be detected. This procedure is performed on each image received from the end node device using HTTP PUT requests. The HTTP PUT method requires that the requested URI message be updated or created, which is enclosed in the body of the PUT message. Thus, if there is a resource in this URI, the message body is considered as a new modified version of this resource. Once the PUT request is received, the service starts scanning the bees and then the Varroa so you can output an updated CSV file containing the number of Varroa mites detected in each photo taken by the end node device. Figure 2 shows in detail the steps of the detection process implemented on the cloud server (online) or in the embedded end-node device (offline).

The following section presents the authors' experimental results using the proposed CNNs combined with computational vision techniques. Figure 3 shows the detection output from each one of the steps that are followed by the author's proposed detection process as illustrated in the flowchart in Figure 2. The images displayed/per detection step have been taken during the Varroa mite detection process validation.

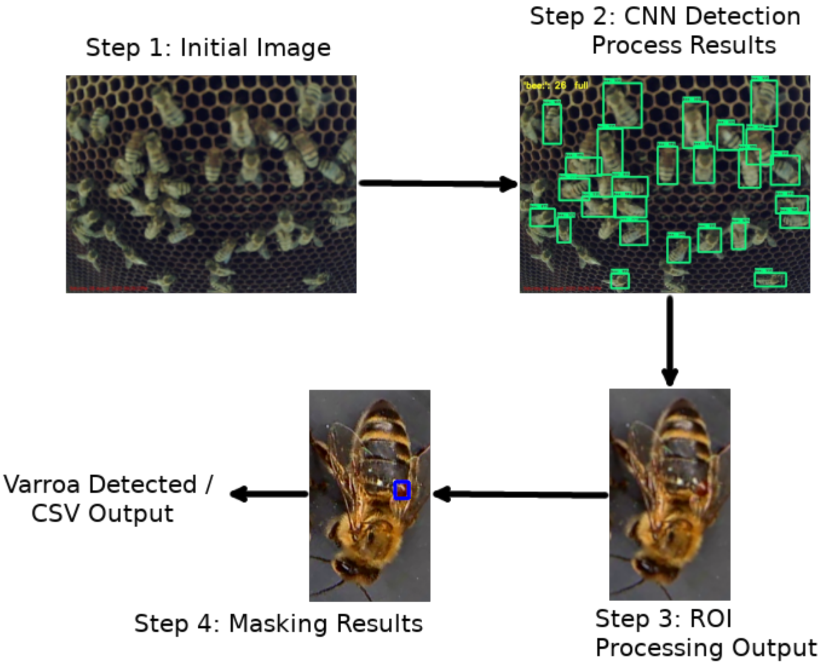
Step 1: Initial Image
Step 2: CNN Detection Process Results
Step 3: ROI Processing Output
Step 4: Masking Results
Varroa Detected / CSV Output

Figure 3. Proposed algorithm step-by-step detection output per flowchart detection process step.

5. Experimental Scenarios and Results

This section presents the experimental scenarios that were implemented using end-node devices to detect bees using a selected CNN (Faster R-CNN), that uses the region proposal network (PRN) to reduce the total image detection times at least 10× times [44], and one of three pre-trained models: MobileNet V2 [55], MobileNet V3 [56], or ResNet-50

FPN [57] (as illustrated in Table 1), and then detect the Varroa mite using computer vision color masking and Hough transformation.

The tests to validate the detection were performed using 100 images from inside the hive that did not contain Varroa and 100 photographs of individual bees from a different set of photos containing Varroa. As a result, the Varroa mite could be detected.

For each test, five different measurements were taken: The time required to load the trained network into the system memory, the total detection time, the average ROI detection time, the mean Average Precision (mAP) obtained from testing the models for IoU = 0.5 and the maximum memory distribution per neural network. In addition, two different measurements have been measured for Varroa detection: The average detection accuracy and the Varroa mite detection time for each bee ROI (detected contour).

In addition, the Detection Accuracy (*DA*) and the Mean Detection Accuracy (*MDA*) metrics have been used (see Equation (1)) to evaluate the proposed Varroa detection process. *DA* and *MDA* metrics are calculated by manually measuring the bees in the test photos and the Varroa mites in each photo with individual bees. The accuracy metrics have been mentioned at [58].

$$DA = \frac{DC}{N}, MDA = \frac{1}{M}\sum_{i=1}^{M} DA_i \tag{1}$$

$$Er = 1 - \frac{DC}{C} \tag{2}$$

Object detection accuracy is also measured by mean Average Precision (mAP) from Pytorch. That is the average of maximum precision at different detected contours over the real annotated ones. Precision measures prediction accuracy by measuring true positives from all true positive and false negative cases.

The tests were performed on three different systems, a Raspberry Pi Zero W version 2 and two cloud servers. The RPi has a quad-core processor clocked at 1 GHz with 364 MB of RAM, 2 GB of swap memory, and a GNU/Linux 64-bit operating system. The cloud server version 1 has an octa-core processor clocked at 2.6 GHz, 12 GB of RAM, 4 GB of swap memory, and its operating system is Ubuntu Server 18.04. While the cloud server version 2 has a 24-core processor clocked at 3.8 GHz, it has 64 GB of RAM, 60 GB of swap memory, and a Ubuntu 20.04 operating system.

5.1. Scenario I: Detection System Performance Tests

The first experimental scenario includes the results obtained from the three systems separately. During the tests, all trained models (MobileNet V2, MobileNet V3, ResNet) of the Faster R-CNN algorithm were evaluated for their performance. The results of the execution time and the memory usage during the detection of bees are presented in Tables 2–4.

Based on the results obtained from the end node device, the Faster R-CNN ResNet-50 model provides the best network load time in the system memory (30% less than the MobileNet V2 model and 7% less than the MobileNet V3 model). Furthermore, a comparison of load times shows that the Faster R-CNN ResNet-50 model provides a faster loading model than MobileNet. However, the average time to detect bees per photo is more extended than MobileNet models. The most efficient model is the MobileNet V3 (80% faster than the Faster R-CNN ResNet-50 and 25% faster than the MobileNet V2).

Then, the execution time of the detection process for the cloud services and the end-node device are compared to measure the value of the speedup σ. Using the average detection time of each model can compute $\sigma = \frac{T_8}{T_{24}}$ as the average detection time using CNN, which is conducted using parallel tasks utilizing all available cores of each system. Therefore, for the MobileNet V2 and V3 models, similar speedups are observed; specifically, the speedups of the models are $\sigma = 3.5$ and $\sigma = 3.4$, when moving from n = 8 to n = 24 cores, respectively. However, for the ResNet-50 FPN model, it provides a greater speedup equal to $\sigma = 6.4$. This is also shown at Tables 3 and 4. By the time each subsystem detects bees in each photo, the ResNet-50 FPN model is 88% faster in cloud server version 2 than version 1 (version 2 has three times more cores than version 1).

Table 2. Experimental Scenario I; end node device operating in offline mode (4 core-ARM CPU) detection time and memory usage.

CNN Models	Load Time (s)	Testing Detection Time (s)	Detection Average Time per Image (s)	Total Time (s)	Memory Usage (MB)
MobileNet V2	125.37	10,848.27	104.56	10,973.63	62.6
MobileNet V3	94.4	8243.58	78.72	8837.98	66.4
ResNet-50 FPN	88.19	38,992.5	385.97	39,080.7	74.2

Table 3. Experimental Scenario I; cloud server version I (octa-core system) detection time and memory usage.

CNN Models	Load Time (s)	Testing Detection Time (s)	Detection Average Time per Image (s)	Total Time (s)	Memory Usage (MB)
MobileNet V2	3.786	265.374	2.33	269.16	81.5
MobileNet V3	2.037	205.216	1.808	207.25	80.2
ResNet-50 FPN	1.514	681.78	6.555	683.29	81.4

Table 4. Experimental Scenario I; cloud server version II (24-core system) detection time and memory usage.

CNN Models	Load Time (s)	Testing Detection Time (s)	Detection Average Time per Image (s)	Total Time (s)	Memory Usage (MB)
MobileNet V2	2.614	74.78	0.47	77.4	13.4
MobileNet V3	1.92	59.32	0.37	61.24	13.1
ResNet-50 FPN	2.45	106.77	0.831	109.23	13.3

5.2. Scenario II: Detection Algorithm Accuracy Tests

Scenario II focuses on the precision results obtained from using the generated neural networks on all three devices operating either in online or offline mode. The measurements used are the Mean Detection Accuracy (MDA) from Equation (1), the mean Average

Precision (mAP) calculated by Pytorch, and the Success Frequency (S.F.) from Equation (3), based on MDA and from Equation (4), based on mAP, as proposed in [58]. The results from the three systems are presented in Table 5.

Table 5. Mean Detection Accuracy (human bee counting verification) over models' mAP.

CNN Models	Mean Detection Accuracy (MDA)	CNN Model mAP	End Node Device SF	Cloud Server Version 1 SF	Cloud Server Version 2 SF
Mobilenet V2	0.887	0.481	0.003	0.17	0.28
Mobilenet V3	0.677	0.496	0.003	0.14	0.29
ResNet-50 FPN	0.821	0.467	0.001	0.1	0.25

Using the metric S.F, the authors can evaluate which of the systems consumes the most energy in the process of detecting bees. Looking at the values in Table 5 and comparing the columns of S.F., it turns out that Cloud Server Version 2 consumes less power for crawling than the other two systems, since the energy needed for crawling is 99% less than the energy required by the End Node Device and 40–60% less than needed by Cloud Server Version 1.

$$SF = \frac{MDA}{T_{load} + T_{detect}} \tag{3}$$

$$SF_{mAP} = \frac{mAP}{T_{load} + T_{detect}} \tag{4}$$

The following scenario describes the results in detecting the Varroa mite, which is the last step in the proposed Varroa detection procedure as illustrated in Figure 2 (detection algorithm's steps and output).

5.3. Scenario III: Evaluation of Varroa Mite Detection Step

In this scenario, 200 photos were used to detect the Varroa mite on bee contours derived from the previous object detection step of the CNN models. This scenario aims to test the accuracy and precision of the Varroa mite detection process on bee objects. One hundred of the detected contours depicted bees with Varroa, while the rest depicted bees without Varroa. Upon the application of the HSV masking and Hough transformation detection step for Varroa mites on bee objects, the measurements in Table 6 were obtained. Table 6 shows the accuracy and precision calculated using the following Equation (5) for accuracy and precision accordingly.

$$\begin{gathered} Acc = \frac{TP + TN}{TP + FP + FN + TN},\ Prec = \frac{TP}{TP + FP}, \\ Recall = \frac{TP}{TP + FN},\ F1\_Score = \frac{2 * Recall * Prec}{Recall + Prec} \end{gathered} \tag{5}$$

The variables used for the validation process are: *TP* (True Positive) describes the number of photos in which the mite is present and detected successfully, *FN* (False Negative) is the number of photos that had Varroa but were not detected, *FP* (False Positive) is the number of images that did not have Varroa but was detected, and finally the variable *TN* (True Negative) is the number of images that did not have the Varroa mite and was not detected.

Table 6. Accuracy and Precision of image processing step on CNN bee-detected contours using HSV color masking and Hough transformation.

Class Test	Predicted Varroa	Predict No Varroa	Accuracy (%)	Precision (%)
Actual Varroa	TP = 64	FN = 36	77	86
Actual No Varroa	FP = 10	TN = 90		

In addition, using the measurements of Table 6 along with the measurements of scenario II (Table 5), the total accuracy of the whole Varroa detection process can be inferred through the formula: $Total_{Acc} = Acc * MDA$. In this way, it can be determined which CNN, together with the use of image processing, can give the best results for the detection of Varroa. This is because the accuracy rates of Table 6 depend to a large extent on the number of images that go through the process of detecting the Varroa mite. The MobileNet V2 model has the best total accuracy, which corresponds to 68% (8% more than the ResNet-50 FPN and 24% more than the MobileNet V3).

It should be commented at this point that the authors have not used the Recall and F1-Score metrics. The recall metric refers to the sensitivity percentage of the varroa mite detection algorithm. The degree of sensitivity depends directly on the size of the masked color space to recognize the Varroa color hue. This space also depends on the brightness and sharpness of the photos that go through this process. For this reason, the F1-Score for the calculation of which recall in part has not been taken into account. Instead, only precision and accuracy metrics have been used.

Using the same set of photos for all system modes of operation (online version 1, online version 2, offline end-node device), there were no differences in the execution times of the detection process, since, in all three cases, the average detection time is close to 4ms. Moreover, there was no change in the size of the memory used by the three device cases during the Varroa detection step, which is close to 0.7 MB.

5.4. System Cross Comparison with Existing Literature Solutions

Summarizing the existing camera-based technological solutions, as presented in the bibliography, to detect Varroa mite, are presented in Table 7. Table 7 characterizes the proposed system capabilities based on their camera facing setup (inside the brood box facing bee frames or at the beehive door openings), maximum precision recorded, mean detection time per photo, and whether they are capable of online and offline operations. That is, the CNN is implemented in the cloud only and requires end-node devices capable of transmuting images only, or it also supports an embedded CNN implementation, having end-nodes that include the CNN algorithms, and no Internet connectivity is required.

From Table 7 it is obvious that most of the systems include a camera setup facing the beehive door openings. This is not the case for the early detection of Varroa mite since it grows inside the brood-box frames, making most of the existing propositions incapable of early detection. In addition, only the author's proposed end-node devices and Chazette's [34] may offer offline operation embedded at the end-node devices. Mrozek's [33] and Bilik's [36] implementations are mostly cloud-based, with the Bilik's algorithm to also utilize also GPU cores for the detection process.

Table 7. Cross comparison of camera-based detection systems for the detection of Varroosis

Proposed Solutions	Camera Facing Frames	Max Detection Precision (%)	Mean Detection Time (ms)	Offline Operation
Mrozek et al. [33]	No	36	270	No
Chazette et al. [34]	No	72	2500	Yes
Bilik et al. [36]	-	87	0.05	No
Authors' system	Yes	86	104,560 (Offline), 470 (Online)	Yes

The authors' end-node detection algorithm has similar precision results to the Yolov5 algorithm proposed in [36]. The authors also express serious doubts about the mean detection time of Chazette's CNN implementation, executed in an embedded system that can achieve more than 72% precision. Accuracy measurements are not considered, since most existing systems use the F1 score as an evaluation metric. For that reason, the authors also assume that F1 score values represent mainly precision values of a constant recall value for all system cases examined.

According to Table 7, the authors' proposed online detection time is about 470 ms. That is two times more than Mrozek's solution [33]. Nevertheless, the maximum precision achieved by Mrozek's model is 36%, which is 2.3 times less than the precision achieved by our model. Comparing the offline operation of our model and Chazette's [34], the authors also mention that Chazette's implementation is not part of an embedded solution rather than an experimental proof of concept run on a test-bed PC.

The authors have also preliminarily experimented with the EfficientNet model [59], which is a lightweight model implementation similar to MobileNet V2 and V3 models and has similar accuracy results to the MobileNet V2 model. Nevertheless, loading the EfficientNet trained Neural Network into the system's memory takes 92% more time than the MobileNet V2 model. In addition, the total time required to detect bees using EfficientNet is 20% longer than the time required by MobileNet V2. Taking into account the above excessive model load time, the authors have chosen not to include EfficientNet in this study. The authors set as future work a further reduction of their model offline detection time (which also includes model load time).

6. Conclusions

This paper presents a novel Varroa mite detection system that uses cameras and deep learning techniques embedded into the beehive for early detection. The authors present both their system and their detection process. The proposed CNN detection algorithm can be incorporated either in the cloud or at the end-node system devices, modifying the system's architecture and operation accordingly.

The authors have experimented with using their system's detection process both online (performing detection process as a cloud-service) and offline (end-node device includes the detection algorithm and provides textual detection output over BLE). The experimental results have demonstrated that the detection process over the cloud is at least 3–4 times faster than if the algorithm is incorporated as a service at the end-node embedded device. The online mode also offers the advantage of easily changing the CNN model with a newly trained one. The disadvantage of the online mode includes communication providers' imagery data uploading costs using NBIoT or LTE Cat-M1 transceivers. On the other hand, in offline mode, the mean detection time per image of 104 s and the CNN model load time of 125 s (MobileNet V2 model) at the embedded quad-core ARM MCU is relatively 40×–80× times more than the time required for online detection. This detection time interval is significantly less energy efficient since the end-node device is battery operated and requires minimum detections to be performed either once per hour or once per day.

The authors' detection algorithm experimentation focused on using different pre-trained models (MobileNet V2, MobileNet V3, ResNet-50), setting the MobileNet V2 model as the most accurate one in the Faster R-CNN algorithm used the authors' trained network. Finally, the authors experimented with their Varroa mite detection process and provided the evaluation of their detection algorithm's accuracy and precision in cases of actual Varroa incidents to 77% and 86% accordingly.

The authors set for future work the extensive evaluation of their proposed system towards Varroa mite outbursts and the extension of their algorithms accuracy, minimizing offline detection time and further classification of the severity of the Varroa incidents detected.

Author Contributions: Conceptualization, S.K.; methodology, S.K.; software, G.V.; validation, G.V. and A.M.; formal analysis, S.K. and G.V.; investigation, G.V. and A.M.; resources, S.K.; data curation, G.V.; writing—original draft preparation, G.V.; writing—review and editing, S.K.; visualization, G.V. and A.M.; supervision, S.K.; project administration, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research has been supported by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH–CREATE–INNOVATE (project code: T1EDK-2402).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application Interface
BLE	Bluetooth Low Energy
FFT	Frequency Fourier Transformation
GPIO	General Purpose Input Output
CCD	Colony collapse disorder
MDA	Mean Detection Accuracy
MFCC	Mel-frequency cepstrum coefficients
MEL	Melody-Logarithmic scale spectrogram
I2C	Inter-Integrated Circuit
CNN	Convolutional Neural Network
NTP	Network Time Protocol
RTC	Real Time Clock
ROI	Regions Of Interest
WSN	Wireless Sensor Network

References

1. Martinello, M.; Manzinello, C.; Borin, A.; Avram, L.E.; Dainese, N.; Giuliato, I.; Gallina, A.; Mutinelli, F. A Survey from 2015 to 2019 to Investigate the Occurrence of Pesticide Residues in Dead Honeybees and Other Matrices Related to Honeybee Mortality Incidents in Italy. *Diversity* **2020**, *12*, 15. [CrossRef]
2. Oddie, M.; Büchler, R.; Dahle, B.; Kovacic, M.; Le Conte, Y.; Locke, B.; de Miranda, J.R.; Mondet, F.; Neumann, P. Rapid parallel evolution overcomes global honey bee parasite. *Sci. Rep.* **2018**, *8*, 7704. [CrossRef] [PubMed]
3. Flores, J.M.; Gamiz, V.; Jimenez-Marin, A.; Flores-Cortes, A.; Gil-Lebrero, S.; Garrido, J.J.; Hernando, M.D. Impact of Varroa destructor and associated pathologies on the colony collapse disorder affecting honey bees. *Res. Vet. Sci.* **2021**, *135*, 85–95. [CrossRef] [PubMed]
4. Dineva, K.; Atanasova, T. ICT-Based Beekeeping Using IoT and Machine Learning. In Proceedings of the Distributed Computer and Communication Networks, Moscow, Russia, 17–21 September 2018; Springer International Publishing: Cham, Switzerland, 2018; pp. 132–143.
5. Olate-Olave, V.R.; Verde, M.; Vallejos, L.; Perez Raymonda, L.; Cortese, M.C.; Doorn, M. Bee Health and Productivity in Apis mellifera, a Consequence of Multiple Factors. *Vet. Sci.* **2021**, *8*, 76. [CrossRef]

6. Rafael Braga, A.; Gomes, D.G.; Rogers, R.; Hassler, E.E.; Freitas, B.M.; Cazier, J.A. A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies. *Comput. Electron. Agric.* **2020**, *169*, 105161. [CrossRef]
7. De la Mora, A.; Emsen, B.; Morfin, N.; Borges, D.; Eccles, L.; Kelly, P.G.; Goodwin, P.H.; Guzman-Novoa, E. Selective Breeding for Low and High Varroa destructor Growth in Honey Bee (Apis mellifera) Colonies: Initial Results of Two Generations. *Insects* **2020**, *11*, 864. [CrossRef]
8. Le Conte, Y.; Arnold, G.; Desenfant, P. Influence of Brood Temperature and Hygrometry Variations on the Development of the Honey Bee Ectoparasite Varroa jacobsoni (Mesostigmata: Varroidae). *Environ. Entomol.* **1990**, *19*, 1780–1785. [CrossRef]
9. Kraus, B.; Velthuis, H.H.W. High Humidity in the Honey Bee (Apis mellifera L.) Brood Nest Limits Reproduction of the Parasitic Mite Varroa jacobsoni Oud. *Naturwissenschaften* **1997**, *84*, 217–218. [CrossRef]
10. Zacepins, A.; Kviesis, A.; Pecka, A.; Osadcuks, V. Development of Internet of Things concept for Precision Beekeeping. In Proceedings of the 18th International Carpathian Control Conference (ICCC), Sinaia, Romania, 28–31 May 2017; Volume 1, pp. 23–27. [CrossRef]
11. Meikle, W.G.; Weiss, M.; Maes, P.W.; Fitz, W.; Snyder, L.A.; Sheehan, T.; Mott, B.M.; Anderson, K.E. Internal hive temperature as a means of monitoring honey bee colony health in a migratory beekeeping operation before and during winter. *Apidologie* **2017**, *48*, 666–680. [CrossRef]
12. Kady, C.; Chedid, A.M.; Kortbawi, I.; Yaacoub, C.; Akl, A.; Daclin, N.; Trousset, F.; Pfister, F.; Zacharewicz, G. IoT-Driven Workflows for Risk Management and Control of Beehives. *Diversity* **2021**, *13*, 296. [CrossRef]
13. Kontogiannis, S. An Internet of Things-Based Low-Power Integrated Beekeeping Safety and Conditions Monitoring System. *Inventions* **2019**, *4*, 52. [CrossRef]
14. Mekala, M.S.; Viswanathan, P. A Survey: Smart agriculture IoT with cloud computing. In Proceedings of the 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Vellore, India, 10–12 August 2017; Volume 1, pp. 1–7. [CrossRef]
15. Zabasta, A.; Kunicina, N.; Kondratjevs, K.; Ribickis, L. IoT Approach Application for Development of Autonomous Beekeeping System. In Proceedings of the 2019 International Conference in Engineering Applications (ICEA), Sao Miguel, Portugal, 8–11 July 2019; Volume 1, pp. 1–6. [CrossRef]
16. Gil-Lebrero, S.; Quiles-Latorre, F.J.; Ortiz-Lopez, M.; Sanchez-Ruiz, V.; Gamiz-Lopez, V.; Luna-Rodriguez, J.J. Honey Bee Colonies Remote Monitoring System. *Sensors* **2017**, *17*, 55. [CrossRef]
17. Zacepins, A.; Kviesis, A.; Stalidzans, E.; Liepniece, M.; Meitalovs, J. Remote detection of the swarming of honey bee colonies by single-point temperature monitoring. *Biosyst. Eng.* **2016**, *148*, 76–80. [CrossRef]
18. Ntawuzumunsi, E.; Kumaran, S.; Sibomana, L. Self-Powered Smart Beehive Monitoring and Control System (SBMaCS). *Sensors* **2021**, *21*, 3522. [CrossRef]
19. Nolasco, I.; Terenzi, A.; Cecchi, S.; Orcioni, S.; Bear, H.L.; Benetos, E. Audio-based Identification of Beehive States. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; Volume 1, pp. 8256–8260. [CrossRef]
20. Simic, M.; Starcevic, V.; Kezić, N.; Babic, Z. Simple and Low-Cost Electronic System for Honey Bee Counting. In Proceedings of the 28th International Electrotechnical and Computer Science Conference, Portorož, Slovenia, 23–24 September 2019.
21. Odemer, R. Approaches, challenges and recent advances in automated bee counting devices: A review. *Ann. Appl. Biol.* **2022**, *180*, 73–89. [CrossRef]
22. Liew, L.H.; Lee, B.Y.; Chan, M. Cell detection for bee comb images using Circular Hough Transformation. In Proceedings of the nternational Conference on Science and Social Research (CSSR), Kuala Lumpur, Malaysia, 5–7 December 2010; pp. 191–195. [CrossRef]
23. Baptiste, M.; Ekszterowicz, G.; Laurent, J.; Rival, M.; Pfister, F. Bee Hive Traffic Monitoring by Tracking Bee Flight Paths. 2018. Available online: https://hal.archives-ouvertes.fr/hal-01940300/document (accessed on 20 September 2019).
24. Gregorc, A.; Sampson, B. Diagnosis of Varroa Mite (Varroa destructor) and Sustainable Control in Honey Bee (Apis mellifera) Colonies—A Review. *Diversity* **2019**, *11*, 243. [CrossRef]
25. Prost, P.J.; Medori, P. *Apiculture*, 6th ed.; Intercept Ltd.: Mauritius, France, 1994.
26. de Ruijter, A.; v.d. Eijnde, J. Detection of Varroa Mite in the Netherlands Using Tobacco Smoke. *Bee World* **1984**, *65*, 151–154. doi: 10.1080/0005772X.1984.11098803. [CrossRef]
27. Clement, H. *Modern Beekeeping*; Psichalos Publications: Athens, Greece, 2017.
28. Ontario Ministry of Agriculture, Food and Rural Affairs. Varroa Mite—Sampling and Monitoring Infestation Levels. 2019. Available online: http://www.omafra.gov.on.ca/english/food/inspection/bees/varroa-sampling.htm (accessed on 13 March 2019).
29. Szczurek, A.; Maciejewska, M.; Zajiczek, Z.; Bak, B.; Wilk, J.; Wilde, J.; Siuda, M. The Effectiveness of Varroa destructor Infestation Classification Using an E-Nose Depending on the Time of Day. *Sensors* **2020**, *20*, 2532. [CrossRef]
30. Edwards Murphy, F.; Popovici, E.; Whelan, P.; Magno, M. Development of an heterogeneous wireless sensor network for instrumentation and analysis of beehives. In Proceedings of the 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, Pisa, Italy, 11–14 May 2015; pp. 346–351. [CrossRef]

31. Bellos, C.V.; Fyraridis, A.; Stergios, G.S.; Stefanou, K.A.; Kontogiannis, S. A Quality and disease control system for beekeeping. In Proceedings of the 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Preveza, Greece, 24–26 September 2021; pp. 1–4. [CrossRef]
32. Sevin, S.; Tutun, H.; Mutlu, S. Detection of Varroa mites from honey bee hives by smart technology Var-Gor: A hive monitoring and image processing device. *Turk. J. Vet. Anim. Sci.* **2021**, *45*, 487–491. [CrossRef]
33. Mrozek, D.; Gorny, R.; Wachowicz, A.; Malysiak-Mrozek, B. Edge-Based Detection of Varroosis in Beehives with IoT Devices with Embedded and TPU-Accelerated Machine Learning. *Appl. Sci.* **2021**, *11*, 11078. [CrossRef]
34. Chazette, L.; Becker, M.; Szczerbicka, H. Basic algorithms for bee hive monitoring and laser-based mite control. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–8. [CrossRef]
35. Kaur, M. A CNN-Based Identification of Honeybees' Infection Using Augmentation. Master's Thesis, Unitec Institute of Technology, Auckland, New Zealand, 2021.
36. Bilik, S.; Kratochvila, L.; Ligocki, A.; Bostik, O.; Zemcik, T.; Hybl, M.; Horak, K.; Zalud, L. Visual Diagnosis of the Varroa Destructor Parasitic Mite in Honeybees Using Object Detector Techniques. *Sensors* **2021**, *21*, 2764. [CrossRef] [PubMed]
37. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
38. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Proceedings of the Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
39. Gomez, K.; Riggio, R.; Rasheed, T.; Granelli, F. Analysing the energy consumption behaviour of WiFi networks. In Proceedings of the IEEE Online Conference on Green Communications, Piscataway, NJ, USA, 26–29 September 2011; Volume 1, pp. 98–104. [CrossRef]
40. Yang, J.; Poellabauer, C.; Mitra, P.; Neubecker, C. Beyond beaconing: Emerging applications and challenges of BLE. *Ad Hoc Netw.* **2020**, *97*, 102015. [CrossRef]
41. Siva, J.; Yang, J.; Poellabauer, C. Connection-less BLE Performance Evaluation on Smartphones. *Procedia Comput. Sci.* **2019**, *155*, 51–58. [CrossRef]
42. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [CrossRef]
43. Ahmed, A. Pre-trained CNNs Models for Content based Image Retrieval. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 2021. [CrossRef]
44. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497.
45. Shahi, T.B.; Sitaula, C.; Neupane, A.; Guo, W. Fruit classification using attention-based MobileNetV2 for industrial applications. *PLoS ONE* **2022**, *17*, e0264586. [CrossRef]
46. Wang, W.; Yang, Y.; Wang, X.; Wang, W.; Li, J. Development of convolutional neural network and its application in image classification: A survey. *Optical Eng.* **2019**, *58*, 1–19. [CrossRef]
47. Bharati, P.; Pramanik, A. Deep Learning Techniques-R-CNN to Mask R-CNN: A Survey. In *Proceedings of the Computational Intelligence in Pattern Recognition*; Springer: Singapore, 2020; pp. 657–668.
48. Tzudalin, D. LabelImg Is a Graphical Image Annotation Tool and Label Object Bounding Boxes in Images. 2016 Available online: https://github.com/tzutalin/labelImg (accessed on 18 September 2019).
49. Labelbox Company. Labelbox: The Leading Training Data Platform for Data Labeling. Available online: https://labelbox.com (accessed on 5 June 2021).
50. Image Annotation Tool. Available online: https://github.com/alexklaeser/imgAnnotation (accessed on 3 June 2021).
51. Team, O. Computer Vision Annotation Tool (CVAT). 2021. Available online: https://github.com/openvinotoolkit/cvat (accessed on 15 June 2021).
52. Lim, B.; Son, S.; Kim, H.; Nah, S.; Lee, K.M. Enhanced Deep Residual Networks for Single Image Super-Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; Volume 1, pp. 1132–1140. [CrossRef]
53. Bradski, G. The OpenCV Library. *Dr. Dobb's J. Softw. Tools* **2000**, *120*, 122–125.
54. Models and Pre-Trained Weights—Torchvision 0.12 Documentation. Available online: https://pytorch.org/vision/stable/models.html (accessed on 9 January 2022).
55. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv* **2019**, arXiv:1801.04381.
56. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. *arXiv* **2019**, arXiv:1905.02244.
57. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.

58. Voudiotis, G.; Kontogiannis, S.; Pikridas, C. Proposed Smart Monitoring System for the Detection of Bee Swarming. *Inventions* **2021**, *6*, 87. [CrossRef]
59. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 6105–6114.