Int. J. Appl. Math. Comput. Sci., 2022, Vol. 32, No. 3, 355–369
DOI: 10.34768/amcs-2022-0026

EDGE COMPUTING IN IoT–ENABLED HONEYBEE MONITORING FOR THE DETECTION OF *VARROA DESTRUCTOR*

ANNA WACHOWICZ [a], JAKUB PYTLIK [a], BOŻENA MAŁYSIAK-MROZEK [b], KRZYSZTOF TOKARZ [c], DARIUSZ MROZEK [a,*]

[a]Department of Applied Informatics
Silesian University of Technology
ul. Akademicka 16, 44-100 Gliwice, Poland
e-mail: dariusz.mrozek@polsl.pl

[b]Department of Distributed Systems and Informatic Devices
Silesian University of Technology
ul. Akademicka 16, 44-100 Gliwice, Poland

[c]Department of Graphics, Computer Vision, and Digital Systems
Silesian University of Technology
ul. Akademicka 16, 44-100 Gliwice, Poland

Among many important functions, bees play a key role in food production. Unfortunately, worldwide bee populations have been decreasing since 2007. One reason for the decrease of adult worker bees is varroosis, a parasitic disease caused by the *Varroa destructor* (*V. destructor*) mite. Varroosis can be quickly eliminated from beehives once detected. However, this requires them to be monitored continuously during periods of bee activity to ensure that *V. destructor* mites are detected before they spread and infest the entire beehive. To this end, the use of Internet of things (IoT) devices can significantly increase detection speed. Comprehensive solutions are required that can cover entire apiaries and prevent the disease from spreading between hives and apiaries. In this paper, we present a solution for global monitoring of apiaries and the detection of *V. destructor* mites in beehives. Our solution captures and processes video streams from camera-based IoT devices, analyzes those streams using edge computing, and constructs a global collection of cases within the cloud. We have designed an IoT device that monitors bees and detects *V. destructor* infestation via video stream analysis on a GPU-accelerated Nvidia Jetson Nano. Experimental results show that the detection process can be run in real time while maintaining similar efficacy to alternative approaches.

Keywords: Internet of things, IoT, *Varroa destructor*, precision beekeeping, machine learning, cloud, image processing, edge devices.

1. Introduction

Bees are among the most important creatures in the animal kingdom. They are responsible for the pollination of approximately 85% of flowers (Zacepins *et al.*, 2017a) and 75% of crops, which account for more than one third of global food production (Debauche *et al.*, 2018). Unfortunately, yearly counts of bee population have inexplicably dropped since 2007 in the United States, Canada, Europe, and Asia. The many possible causes of colony collapse disorder—a rapid drop in worker bee numbers (vanEngelsdorp *et al.*, 2017; 2007)—include pesticides, antibiotics, starvation, malnutrition, and pathogens (van der Sluijs *et al.*, 2013; Cornman *et al.*, 2012; Barron, 2015). Guzmán-Novoa *et al.* (2010) discovered that the *Varroa destructor* (*V. destructor*) mite is the leading cause of morbidity factors. This finding was confirmed by Dineva and Atanasova (2018).

V. destructor mites live on adult bees and bee brood (larvae). The mites cause varroosis (*Varroosis apium*),

*Corresponding author

the world's most destructive honey bee disease, which inflicts substantially greater damage and incurs higher economic costs than all other known apicultural diseases. *V. destructor* mites are a leading cause of bee mortality throughout Europe and other continents. Varroosis has a non-uniform disease pattern, as clinical symptoms are determined by both the rate of infestation and secondary infections (Boecking and Genersch, 2008).

V. destructor mites have a diameter of approximately 1.4 mm; individual mites are visible to the naked eye. Using their chelicerea, the mites extract hemolymph from the host bees during the larval, pupal, and adult stages. Principally, this process has two sets of dangerous effects on the health of the bees. Firstly, the loss of hemolymph negatively affects the organ development of the bee, weakening the navigational capabilities, reducing the flying capacity, and shortening the adult life of the bees by up to 68% (Bojanic Rasovic *et al.*, 2018; Schneider and Drescher, 1987). Secondly, while draining hemolymph, the mites spread various types of viral diseases, including Kashmir bee virus, acute bee paralysis virus, and deformed wing virus. Such afflictions decimate bee populations (Chen and Siede, 2007).

The development cycle of the *V. destructor* mite lasts 9–10 days. During this time, each female *V. destructor* lays between three and six eggs within sealed cells of bee brood. Of these eggs, approximately 2/3 will produce female mites and 1/3 will produce male mites. For a *V. destructor* infection among an untreated bee family, several individual mites in the spring will give rise to several thousand mites in the autumn. As a result, the beehive would be unlikely to survive the winter. Bee colonies infested by *V. destructor* mites will develop the parasitic mite syndrome and ultimately die without treatment. A poorly coordinated treatment and the lack of effective monitoring methods result in the widespread recurrence of a bee colony collapse. Further viral infections increase the risk of colony losses. No effective drug exists that can reliably eliminate 100% of mites and hence eradicate varroosis from beehives. Thus, a selective action is required to eliminate varroosis infections while not harming bees. The control of *V. destructor* mites is a regular task for beekeepers. Many methods exist to combat the mites. However, regardless of the method, early detection is a priority.

In this paper, we demonstrate that varroosis can be detected early and monitored in real time by the use of Internet of things (IoT) technologies combined with artificial intelligence (AI). Our proposed solution allows beekeepers to be notified, and provides an automated process that will minimize human interference in beehives. Our system is built on an Nvidia Jetson Nano platform that can record and process video frames, and enables the use of edge analysis with machine learning models.

The main contributions of this paper are as follows:

1. We examine different algorithms for bee identification in video streams (adaptive algorithms with background subtraction and machine learning-based one) and compare them to choose the final one to be implemented in the edge IoT device.

2. We investigate deep learning-based image analysis methods for bee identification and *V. destructor* detection and prove that they are capable of performing their tasks with efficacy comparable to approaches that rely on the analysis of other types of data (e.g., gas analysis).

3. To enable real-time, on-edge detection of varroosis, we investigate a unique combination of multiple parameters, such as the number of skipped frames of the video stream or various camera resolutions. Time performance and quality measures are analyzed to meet this requirement.

4. To support the scalability of a system, we proposed a cloud-based architecture with edge-processing devices that lays a solid background for further extensions on monitoring multiple beehives simultaneously.

5. Finally, with edge-based IoT detection, our approach allows reducing the amount of data sent for analysis in larger data centers.

The remainder of the paper is organized as follows. Section 2 presents the state-of-the-art regarding recent solutions for bee monitoring. Section 3 describes our experimental environment. Section 4 presents the results of the experimental validation. Finally, Section 5 discusses and compares the obtained results with those from the literature, and Section 6 summarizes the paper.

2. Related works

Bee monitoring systems have a long history. Gates (1914) described yearly temperature measurements. Following this, Dunham (1931) proposed an electrical thermocouple to monitor temperature conditions inside hives. Thereafter, multitudinous solutions were developed and described within the scientific literature. IoT solutions are commonly utilized in human healthcare monitoring and diagnosis (Domański *et al.*, 2021; Wojnakowski *et al.*, 2021). Recently, we demonstrated a successful use of wearable IoT devices for the monitoring of patients with arthritis (Mielnik *et al.*, 2019; 2021), the behavior of elderly people (Mrozek *et al.*, 2020a), and rehabilitation progress (Rodak *et al.*, 2022).

Systems for the monitoring of bee health, including IoT solutions, are collectively known as precision beekeeping (or precision apiculture) (Zacepins *et al.*, 2017a; Kviesis and Zacepins, 2015; Zogovic *et al.*, 2017). Such systems also exist for other branches of agriculture, such as fruit growing (Van Goethem *et al.*, 2019). The development of precision beekeeping systems is motivated by the need to monitor specific groups of parameters. Meikle and Holst (2014) review the methods used and the parameters monitored. Bayir and Albayrak (2016) developed a wireless sensor network for monitoring nectar flow based on the temperature and humidity of a bee colony, and the weight of the beehive. As shown by Debauche *et al*. (2018), bee health can also be monitored by IoT technologies and cloud platforms. The authors presented an example architecture for a cloud-based system, but only monitored temperature, humidity, barometric pressure, and light.

Several commercial or scientific solutions—such as ULmonitor (Krzyśka, 2022), wireless apiary monitoring, or beehive monitoring—allow multiple parameters to be measured, including temperature and humidity (Gil-Lebrero *et al*., 2017; Rustia *et al*., 2016), beehive weight (Dasig and Mendez, 2020; Ochoa *et al*., 2019; Zabasta *et al*., 2019; Fitzgerald *et al*., 2015; Zacepins *et al*., 2017b; Braga *et al*., 2019), carbon dioxide levels within the hive (Edwards-Murphy *et al*., 2016), air pressure (Machhamer *et al*., 2020), the frequency of the sound generated by the bees (Chen *et al*., 2020; Zgank, 2021; Balta *et al*., 2017; Bencsik *et al*., 2011; Zgank, 2020), and forager traffic (Machhamer *et al*., 2020). Moreover, certain systems allow video monitoring (Meitalovs *et al*., 2009), including bee counting (Campbell *et al*., 2008; Chen *et al*., 2012). Some systems combine multiple functions. For example, the solution proposed by Babic *et al*. (2016) simultaneously detects and counts honey-carrying bees. Qandour *et al*. (2014) develop a system to analyze bee noises and hence distinguish the behavior of queens, scouts, foragers, and the entire bee family. Beyond monitoring, some systems allow beekeepers to adjust conditions within a beehive when some measured parameters are not within an acceptable range (Kontogiannis, 2019).

Despite the large variety of monitoring solutions, very few focus on the detection of varroosis. Marstaller *et al*. (2019) and König (2019) present complex, energy-efficient systems that use image-based algorithms for bee detection and classification, or bee counting. Both groups of authors intend to extend their solutions with *V. destructor* detection modules; however, the modules are not yet implemented. Chen *et al*. (2020) analyze thermal images to track various viral infections, including *V. destructor* mites, but as of yet have presented no results. Bjerge *et al*. (2019) analyze recorded video sequences to monitor *V. destructor* infestation levels. Schurischuster *et al*. (2016) tested different camera setups for visual detection of infested bees, but do not propose an algorithm for the detection of mites, focusing only on high quality video capture. Schurischuster *et al*. (2018) extract individual bee images, and classify them as mite-infested or not mite-infested using image analysis and machine learning techniques. Elizondo *et al*. (2013) propose a system that can detect *V. destructor* mites within honeybee cells. However, the system is incomplete and cannot collect image data. Further solutions detect varroosis without using IoT methods (Szczurek *et al.*, 2020; 2019). These systems perform a chemical analysis of the air within the beehive, based on the assumption that the presence of *V. destructor* mites influences gas composition.

The above works demonstrate that machine learning methods can be used for the detection of varroosis within beehives. However, to apply such processes on a large scale requires the use of IoT technologies—including AI, cloud computing, and big data (Stefanowski *et al*., 2017)—and the development of dedicated methods to combine such approaches. This paper works towards the creation of such a system, and extends existing solutions by providing the following:

(i) a comprehensive IoT solution for monitoring bees that uses edge-based monitoring and detection, and is capable of acquiring and analyzing a video stream, communicating the results with a cloud-based data center, and notifying beekeepers;

(ii) efficient real-time detection of *V. destructor* mites in apiaries using the Nvidia Jetson Nano hardware accelerator with satisfactory hardware performance, allowing appropriate bee classification algorithms to be selected and embedded within edge IoT devices;

(iii) experimental results validating the efficacy and speed of bee identification using various algorithms, and the detection of *V. destructor* mites using different camera resolutions.

Our prior work with edge-implemented AI systems shows that the embedding of classification models into IoT devices significantly reduces the volume of data transferred to the cloud and the frequency of device-to-cloud communication (Gołosz and Mrozek, 2019; Mrozek *et al*., 2020b). Such an approach may also reduce the energy consumption of the IoT device, which is essential to prolonging operating time (Grzesik and Mrozek, 2021).

3. Edge and cloud computing in bee monitoring

Remote bee monitoring can free beekeepers from frequent beehive inspections. Such monitoring can be possible through the use of edge IoT devices deployed within

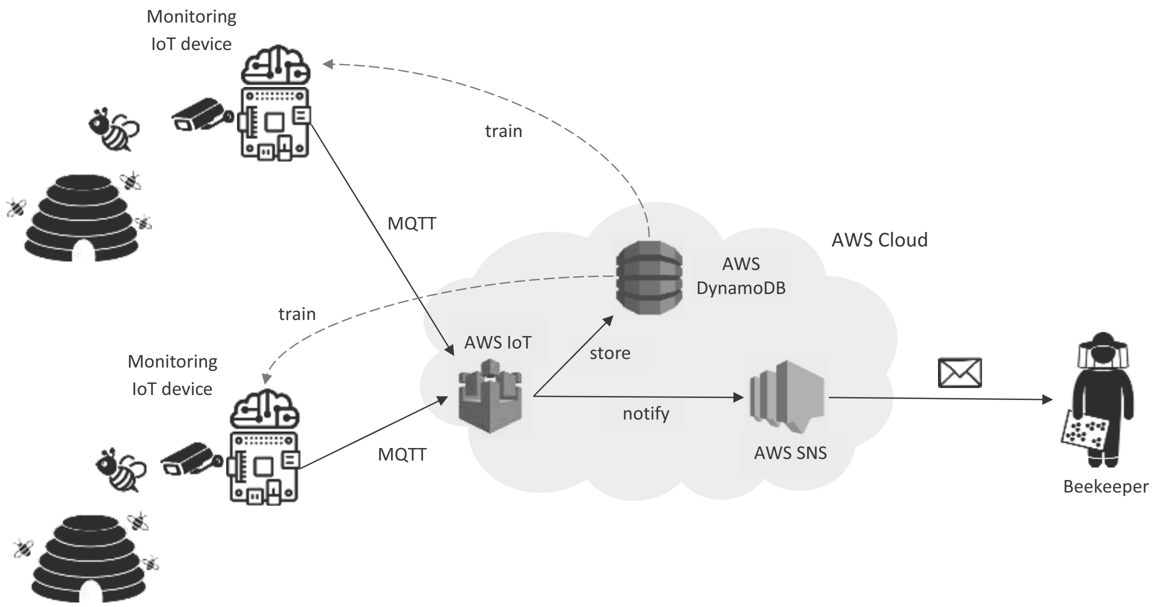


Fig. 1. Architecture of the cloud-based experimental environment for the monitoring of bees and detection of virosis.

an appropriate environment. Edge computing allows storing and processing the data gathered by the monitoring devices closer to where it is produced. As shown in Fig. 1, our environment consists of a monitoring data center and a set of IoT field monitoring devices located near beehives. The IoT edge monitoring devices process video streams, identify bees, and detect *V. destructor* mites using AI methods. The IoT devices deliver data, particularly in the case of varroosis detection, to the cloud data center. The architecture of the data center is shown in Fig. 1, and described in detail in Section 3.1. Section 3.2 discusses the construction of the IoT monitoring devices, while Section 3.3 deals with the video processing and analysis algorithms.

3.1. Cloud environment architecture for large-scale global monitoring. Monitoring multiple beehives within a single apiary may produce useful data that can be employed to build effective AI-based modules for varroosis detection. However, such monitoring can only gather data at a single location. Monitoring on a larger scale, across multiple apiaries, regions, or countries, would both enable early detection of *V. destructor* infestations and allow the spread of varroosis to be tracked regions or across the entire globe. The implementation of such large-scale monitoring requires a highly scalable data center that can permit connections with many IoT monitoring devices and adjust appropriately to the growing numbers. Cloud computing is a natural choice. For our solution we chose to use Amazon Web Services (AWS), a cloud platform that provides relevant services including the ability to connect IoT devices, receive IoT events, store gathered data, and notify users (beekeepers). The AWS cloud platform was used as part of the AWS research grants we obtained from Amazon.

Figure 1 shows the system architecture, including the IoT monitoring devices and the AWS cloud data center. The IoT devices monitor beehives constantly from morning until late evening. The devices can segment the captured video to identify bees, and analyze the segmented images to detect the presence of *V. destructor* mites on the bees. Upon detection of a mite, the corresponding IoT device sends an IoT event to the AWS cloud data center. The IoT event contains important data, including the apiary identifier, the IoT device identifier (associated with a particular beehive), the timestamp, and an image of the afflicted bee. Transmission of the event uses the message queuing telemetry transport (MQTT) protocol, a standard for IoT telemetry.

Introduced in 1999, MQTT remains a popular machine-to-machine communication scheme for IoT networks. Mishra and Kertesz (2020) show the consistent popularity of the protocol when compared to competitors. Released in 2019, the newest version of the protocol (version 5) supports large-scale systems and improves performance in constrained IoT node devices (Banks *et al.*, 07 March 2019). The MQTT protocol is supported by all leading cloud vendors, including AWS, Google Cloud Platform, and Microsoft Azure (Pierleoni *et al.*, 2020). Transmitted IoT events pass through the AWS IoT module, a gateway or access point for IoT devices accessing the cloud. The AWS IoT module relays the events received from each IoT device to the

AWS DynamoDB database and notifies the appropriate beekeeper. AWS DynamoDB is a NoSQL document database which stores the IoT events that contain images of bees afflicted by *V. destructor* mites. Amazon Simple Notification Service (AWS SNS) is used to notify beekeepers by sending text and e-mail messages to predefined recipients. This service is triggered by the AWS IoT component when an infected bee is detected.

Data can be collected more easily from multiple hives at a single location. This enables the construction of superior detection models, which can then be retrained and periodically propagated to the IoT monitoring devices. Given that the analysis of the captured video is performed at the edge, within the IoT monitoring devices, the device-to-cloud data transmission is limited to only those cases when the detection process indicates the presence of a *V. destructor* mite within the beehive. In this way, we can significantly limit the amount of transmitted data. Within the transmitted IoT events, the bee images are of much larger size than the text and numerical data. Consequently, we reduce network access costs, the consumption of power by the IoT devices, and the space occupied in the DynamoDB database.

3.2. IoT monitoring device with the Jetson Nano accelerator. IoT systems lie at the forefront of electronics and information technology development. Thus, new hardware platforms and solutions are frequently produced. At the time of writing, the popular Raspberry PI single-board computer platform is facing strong competition from Odroid, Nvidia Jetson, and Asus Tinker Board, among other solutions. A comparison of Odroid C2 and Raspberry PI 3 presented by Nazir *et al.* (2018) shows similar performance for both platforms. From the benchmark reported by Süzen *et al.* (2020), Nvidia Jetson Nano, equipped with 128 CUDA cores, outperforms Raspberry PI 4 in image classification applications while having comparable power requirements. Such a result is not unexpected, given the experience of Nvidia in developing solutions for computer image generation and processing.

As shown in Fig. 2, our IoT monitoring device consists of a central computing unit, in addition to peripheral devices for information collection, communication, and power maintenance. To enable real-time monitoring of the beehive, the central computing unit is based on the Nvidia Jetson Nano computing board module, equipped with a 1.43 GHz ARM Cortex A57 quad-core processor, an Nvidia Maxwell graphics processor with 128 CUDA cores, and 2 GB of RAM. Nvidia Jetson Nano can run multiple neural networks in parallel to provide image classification, object detection, and segmentation applications. These operations require a power consumption of just 5 W (Nvidia, 32.7.1 Release). By default, Jetson Nano allows peripherals to

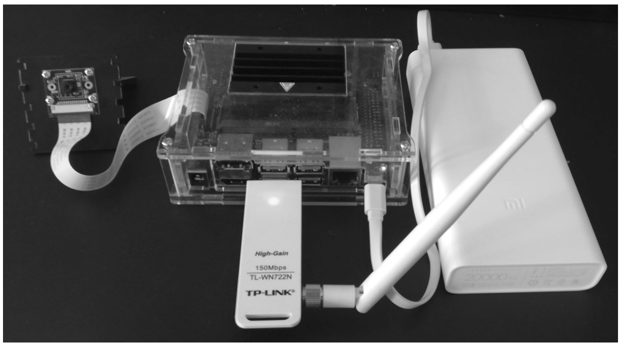


Fig. 2. IoT monitoring device used for bee observation and varroosis detection.

be connected via a USB interface. Thus, communication can be established across multiple Internet interfaces and communication devices. We used the TP-LINK WN722N network card, which enables two-way communication. However, the device still requires an external wireless access point to communicate with the cloud data center via the Internet.

Monitoring the beehive entrance requires the use of a video camera. We employed the Sony IMX219-77 camera module due its hardware compatibility with the Nvidia Jetson Nano module. Moreover, the 8 Mpx matrix of the camera module is sufficient to capture the beehive environment. The camera module is connected to the central computing unit via a camera serial interface. To provide uninterrupted operation of the device for a minimum of 12 hours per day, we used a battery with a capacity of 20,000 mAh.

3.3. Video processing algorithm. The IoT monitoring device operates according to the general workflow shown in Algorithm 1. The algorithm continuously captures the video stream V from the connected camera c (line 4), and processes the video stream using the Nvidia Jetson Nano accelerator (line 5). The algorithm then monitors the cloud buffer and sends an IoT event to the cloud for each bee image b classified as infected (lines 6–8). The IoT event consists of the image b supplemented by the exposition parameters for the image $e(b)$ and the IoT device information d_i retrieved at the outset of the main procedure (line 2). The operations in lines 4–7 are performed in parallel.

The video stream processing algorithm, invoked in line 5 of Algorithm 1 and presented in Algorithm 2, loads the cloud buffer on the IoT device with the images of any afflicted bees that are detected. The algorithm accepts the video stream V as input. It also acquires a handle to the global memory area of the Nvidia Jetson Nano accelerator. The procedure extracts specific frames f from the video stream (line 2) and passes them to the

Algorithm 1. General workflow of the main processing algorithm of the IoT monitoring device.

Require: c {camera}, C_B {cloud buffer}
1: $C_B := \emptyset$
2: $d_i := \mathit{GetDeviceInfo}()$ {incl. device id}
3: **while** true **do**
4:     $V := \mathit{CaptureVideoStream}(c)$
5:     $C_B := \mathit{ProcessVideoStream}(V)$ {Algorithm 2}
6:     **for each** $b \in C_B$ **do**
7:         $\mathit{SendToCloud}(b, e(b), d_i)$
8:     **end for**
9: **end while**

Algorithm 2. General workflow of the video stream processing algorithm executed on the IoT device using the Nvidia Jetson Nano accelerator.

Require: V {video stream}, G {global memory of the GPU accelerator}
1: $B := \emptyset$
2: **for each** $f \in V$ **do**
3:     $G := \mathit{SaveToMemory}(f)$
4:     $B := \mathit{IdentifyBees}(f)$ {Algorithm 3 or 5}
5:     **for each** $b \in B$ **do**
6:         **if** DetectVarroa(b) = true **then**
7:             $C_B := C_B \cup b$
8:         **end if**
9:     **end for**
10: **end for**

Algorithm 3. Preliminary bee identification.

Require: f {frame}
1: $O := \emptyset$
2: $B := \emptyset$;
3: $m := \mathit{BackgrndSub}(f)$
4: $O := \mathit{CreateMapOfObjects}(m, f)$
5: **for each** $o \in O$ **do**
6:     $b := \mathit{ExtractBeeObjects}(o)$ {Algorithm 4}
7:     $B := B \cup b$
8: **end for**
9: **return** B {Extracted outlines of bees/objects}

Algorithm 4. Extraction of bee outlines.

Require: o {image/object}
1: $o' := \mathit{ConvertToGrayScale}(o)$
2: $o' := \mathit{EliminateNoise}(o')$
3: $o' := \mathit{FindBeeOutline}(o')$
4: $b := \mathit{ExtractBeeOutline}(o, o')$
5: **return** b {an extracted bee image}

Algorithm 5. Convolutional neural network bee identification algorithm.

Require: f {a frame separated from video stream}
1: $L := \mathit{CreateCoordinatesWithNN}(f)$
2: **for each** $l \in L$ **do**
3:     $b := \mathit{ExtractBeeObject}(l)$
4: **end for**
5: **return** b {an extracted bee image}

global memory G of the accelerator (line 3). It then executes the bee identification function that returns a set B of bee images detected in frame f (line 4). Next, for each bee image b segmented from the video frame, the algorithm determines the presence of *V. destructor* mites. If infection is found, the image is placed within the cloud buffer C_B (lines 5–9).

Identification of bees with varroosis consists of two phases. During Phase I, the bee identification phase, the bee images are segmented using two algorithms: an adaptive algorithm with background subtraction, and a deep learning algorithm. The former consists of two steps: a preliminary identification process, shown in Algorithm 3, and an exact identification process, shown in Algorithm 4. The identification process begins by subtracting the background of input frame f to create mask m (line 3). We used two adaptive algorithms for background subtraction: k-nearest neighbors (KNN) (Zivkovic and van der Heijden, 2006) and mixture of Gaussian 2 (MOG2) (Zivkovic, 2004). KNN uses recurrence relations for the continuous revision of Gaussians mixtures model parameters, while MOG2 is based on the probability density of Gaussian mixtures. Both frames and generated masks are required to identify bee objects within the frame. During the preliminary identification process, groups of pixels are extracted based on previously generated masks. This step consists of calling *CreateMapOfObject(m, f)* (Algorithm 3, line 4). Next, each object o within the map of objects O is processed by the *ExtractBeeObjects* function (line 6), which finds and selects group of pixels that potentially represent a single bee object. The extraction of appropriate pixels to find bee outlines is performed by Algorithm 4.

The deep learning identification algorithm uses convolutional neural networks (CNNs) and operates as shown in Algorithm 5. By calling the *CreateCoordinatesWithNN* function (line 1), the Nvidia Jetson Nano accelerator finds a list L of bee object regions of interest in the processed frame f. This step uses a CNN provided by Nvidia for the Jetson Nano module. The CNN is a DetectNet model with a sub-layer structure similar to that of the 22-layer GoogLeNet. However, DetectNet differs from GoogLeNet in terms of input, reduce, and output layers. The use of DetectNet enables reduced learning time and better detection results by working with a pre-trained GoogLeNet model. For each region of interest l, Algorithm 5 extracts the bee object by invoking the *ExtractBeeObject* function (lines 2–4).

Once the set of extracted bee objects is created, Phase II of the identification of bees with varroosis begins. During this phase, the video stream processing algorithm (Algorithm 2, line 6) detects the presence of *V. destructor* mites on the extracted bee objects by using a GoogLeNet model trained on images of healthy and infected bees.

4. Experimental results

We verified the ability of our IoT bee monitoring device to detect varroosis in real time. For this purpose, we validated the following components:

- the efficacy of the implemented algorithms in identifying bees within frames of the captured video,
- the time performance of video processing while identifying bees,
- the efficacy of the algorithms used for detecting *V. destructor* mites in extracted bee images,
- the overall time performance of *V. destructor* detection.

Our IoT monitoring device can work continuously for 12 hours on battery power. During this period, it can perform bee monitoring, video processing, bee segmentation (identification), and *V. destructor* detection, while periodically transmitting data to the cloud data center. However, to establish a ground truth for validation of the bee identification and *V. destructor* detection functionality, we used shorter, 29 s video streams. For each frame of the video streams, we manually counted bees, marked regions of interest, and determined the presence of *V. destructor* mites on the extracted bee images.

The efficacy of the algorithms was expressed using several metrics, including the following:

- precision, or positive predicted value (PPV), given by
$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}; \tag{1}$$
- recall (sensitivity), or true positive rate (TPR), given by
$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}; \tag{2}$$
- specificity, or true negative rate (TNR), given by
$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}; \tag{3}$$
and
- F1-score, given by
$$F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}. \tag{4}$$

These metrics use the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates achieved in particular phases of the data analysis. Time performance was measured while running particular algorithms for the same video streams, or while processing extracted bee images.

4.1. Phase I: Bee identification. In the first series of experiments, we evaluated the efficacy and time performance of bee identification when using the adaptive algorithm with background subtraction and the machine learning algorithm with a pre-trained CNN. Both algorithms are used to identify bee objects in individual video frames.

4.1.1. Bee identification efficacy. An effective bee identification process is critical for further analysis and detection of *V. destructor* mites. We tested two adaptive background subtraction algorithms—KNN (Zivkovic and van der Heijden, 2006) and MOG2 (Zivkovic, 2004)—in addition to the DetectNet CNN. The bee identification images were acquired from video streams captured in two resolutions: 960 × 540 px (medium) and 480 × 270 px (low). We tested each algorithm using both medium and low resolutions. For each algorithm and resolution combination, we analyzed 26 selected frames by extracting every 34th frame from the video stream having 890 frames (29 s, 30 fps). The skipping of 33 frames was dictated by performance considerations presented in Section 4.1.2. To prepare the ground truth and calculate the efficacy metrics (precision, sensitivity, and F1 score), we manually counted bee objects within each of the processed frames. For presentation purposes, we renumbered the frames. Frame number 1, as shown in our results, corresponds to an actual frame number of 34; frame number 2, as shown in our results, corresponds to an actual frame number of 64, and so forth.

Figure 3 shows the precision obtained by both adaptive algorithms for both analyzed video streams. The precision fluctuates across different frames, but the curves are similar for both algorithms. High precision is maintained consistently for the low resolution video. For the medium resolution video, the precision fluctuates more strongly, with the KNN algorithm displaying slightly superior performance.

The medium resolution video features a higher level of noise. The background subtraction algorithms are sensitive to noise, and this is reflected in the generated masks. The obtained sensitivity curves are presented in Fig. 4. The video resolution noticeably affects the

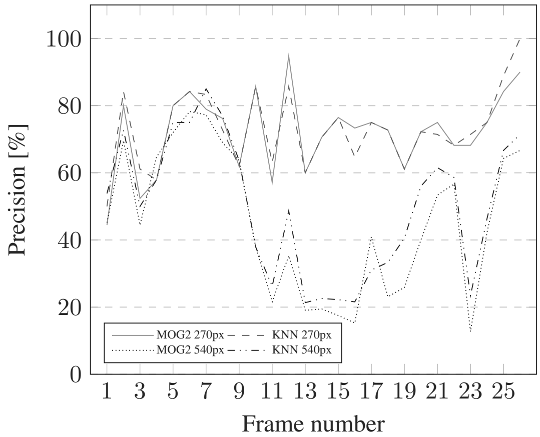
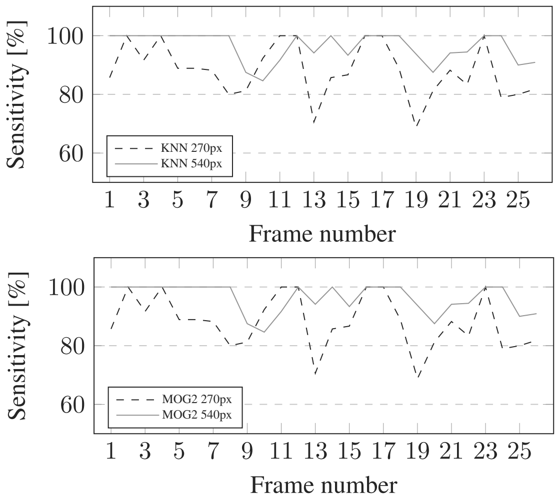


Fig. 3. Bee identification precision when using the KNN and MOG2 algorithms at medium (960 $\times$ 540 px) and low (480 $\times$ 270 px) video resolutions.

a)



b)

Fig. 4. Bee identification sensitivity when using the KNN (a) and MOG2 (b) algorithms at medium (960 $\times$ 540 px) and low (480 $\times$ 270 px) video resolutions.

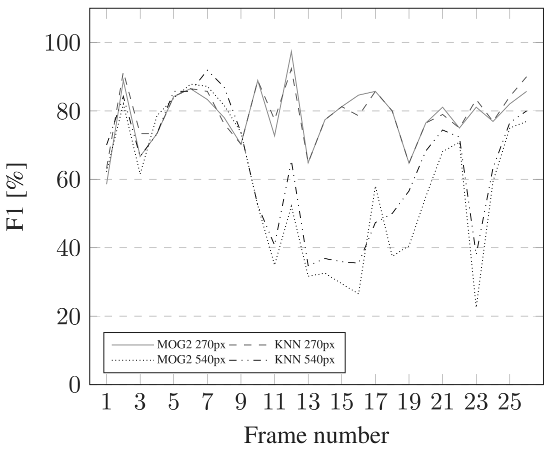


Fig. 5. Bee identification F1 scores when using the KNN and MOG2 algorithms at medium (960 $\times$ 540 px) and low (480 $\times$ 270 px) video resolutions.

sensitivity, with a higher sensitivity of 80% and above obtained for the medium resolution video. However, the sensitivity remains acceptable for the low resolution video, and only infrequently drops to 70%. For both of the algorithms the sensitivity rates are identical when comparing the same video resolution.

Figure 5 presents the bee identification F1 scores for both algorithms. Superior results were produced for the lower resolution video, which featured less noise in the analyzed frames.

We further evaluated bee identification using the DectectNet CNN, provided by Nvidia for the Jetson Nano module. We employed two different video resolutions: 960×540 px (medium) and 1980×1080 px (high). The video streams were captured by monitoring two different beehives. For the medium resolution video, the distance between the camera and the beehive entrance was larger than for the high resolution video. Moreover, the beehives differed in color: the first was predominantly orange and brown while the second was predominantly blue. The DetectNet CNN model was additionally re-trained with 516 frames randomly chosen from the medium resolution video stream. We used the remainder of the video for testing.

Precision, sensitivity, and F1 score results are shown in Figs. 6, 7, and 8, respectively. As shown, superior precision was obtained for the medium resolution video. The sensitivity, critical for bee identification, was similar for both resolutions. The value of each metric was dependent upon the analyzed frame and resolution.

The F1 scores (Fig. 8) reveal that the CNN algorithm is resistant to bias. The scores are similar for both video resolutions. The obtained values are dependent upon both the analyzed frame and the video resolution, but never drop below 80%. Hence, in the majority of cases, the *V. destructor* detection model used in Phase II will be provided with accurately detected bee images.

4.1.2. Bee identification time performance. We verified the overall time performance of the two approaches to bee identification (the KNN adaptive algorithm and the DetectNet CNN). To ensure repeatability of the results, we analyzed a 29 s video stream recorded with a sampling rate of 30 frames per second, for a total of 890 frames. The video stream was prepared in four different resolutions: 1980×1080 px (high resolution), 960×540 px (medium resolution), 480×270 px (low resolution), and 240×135 px (very low resolution).

For the KNN adaptive algorithm, the first 20 frames were used to pre-tune the background subtraction, which was necessary to obtain functionally acceptable masks. Following this, the next frame was selected and processed, 20 further frames were skipped, and then the process was repeated. Video frame processing consisted of three

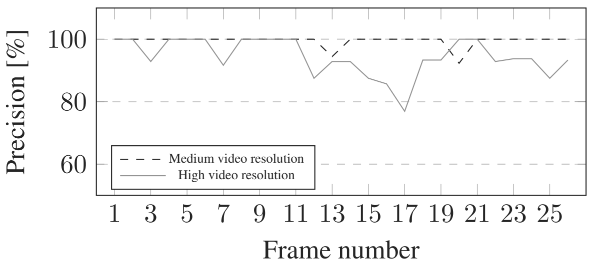


Fig. 6. Bee identification precision when using the DetectNet CNN machine learning algorithm at medium (960 × 540 px) and high (1980 × 1080 px) video resolutions.

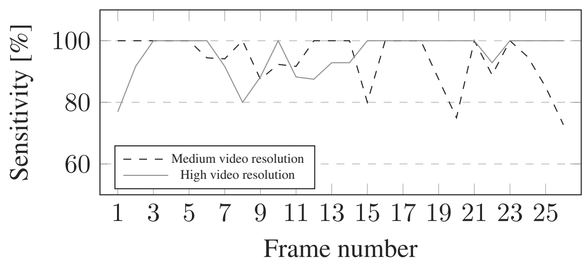


Fig. 7. Bee identification sensitivity when using the DetectNet CNN machine learning algorithm at medium (960 × 540 px) and high (1980 × 1080 px) video resolutions.

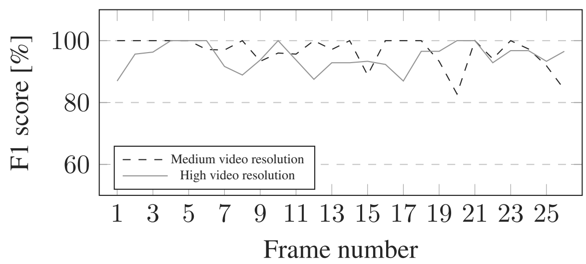


Fig. 8. Bee identification F1 scores when using the DetectNet CNN machine learning algorithm at medium (960 × 540 px) and high (1980 × 1080 px) video resolutions.

steps: frame extraction, preliminary identification, and exact identification. These steps were pipelined within our implementation; the steps overlap during execution but operate on consecutive frames. Time performance was measured from the outset of video processing until the end of the video.

The results are presented in Fig. 9. They show that, for 20 skipped frames, real-time bee identification is only possible when using a very low resolution (240 × 135 px). We investigated that further by verifying the processing time required for different numbers of skipped frames when using a medium resolution (960 × 540 px). The results are shown in Fig. 10. They show that, when using a medium resolution, the video stream can be processed in real time when at least 33 frames are skipped.The highest resolution (1980 × 1080 px) would not enable real-time processing for the adaptive algorithm with a reasonable number of skipped frames, so we did not perform any analysis for it.

We further evaluated the time performance of the DetectNet CNN algorithm for bee identification, at a higher resolution than for the adaptive algorithm. We used videos streams of two different resolutions: 960 × 540 px (medium) and 1980 × 1080 px (high). Both videos had a sampling rate of 30 frames per second. We investigated the processing time required for different numbers of skipped frames. The results are presented in Fig. 11. They show that the DetectNet CNN algorithm can process the medium resolution video stream in real time when skipping 12 frames. When skipping 10 frames, the processing time marginally exceeds the duration of the video. Although the DetectNet CNN algorithm is faster than the adaptive one, it does not allow real-time video processing at high resolution, even when skipping 33 frames.

4.2. Phase II: Varroosis detection.

In the second series of experiments we evaluated the efficacy and time performance of varroosis detection. The *V. destructor* detection model uses the re-trained GoogLeNet CNN. We employed the Caffe deep learning environment to simplify switching between CPU and GPU processing on Nvidia Jetson Nano. Moreover, we optimized the image analysis process via neural network conversion and calibrated the detection model for higher accuracy using TensorRT. Given that only a small number of images showed bees afflicted by *V. destructor*, we used augmentation techniques to increase the number of images of infected bees. In this manner, we obtained 4,800 images, half of which showed healthy bees, and half–infected bees. From this data set we assigned 50 images—25 class N (negative) images of healthy bees and 25 class P (positive) images of infected bees—to the testing set. The remainder of the images formed the training set. The training process took place over 13 epochs.

4.2.1. Varroosis detection efficacy.

We used precision, sensitivity, accuracy, and specificity to analyze the varroosis detection efficacy. The confusion matrix for varroosis detection is presented in Table 1. The matrix shows that 40 out of 50 bee objects were correctly classified. Table 2 shows the precision, sensitivity, accuracy, and specificity rates calculated based on the confusion matrix.

The sensitivity value (68%) indicates that the classifier tends to classify some of the infected bees as healthy. Conversely, the precision (89%) and specificity (92%) values indicate that healthy bees are usually

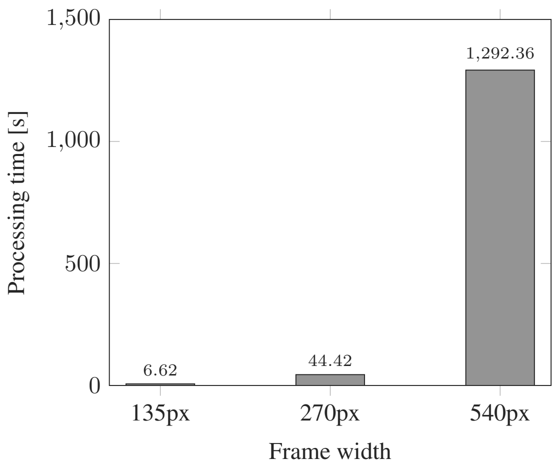


Fig. 9. Bee identification execution time when using the KNN adaptive algorithm at very low (240 × 135 px), low (480 × 270 px), and medium (960 × 540 px) video resolutions.

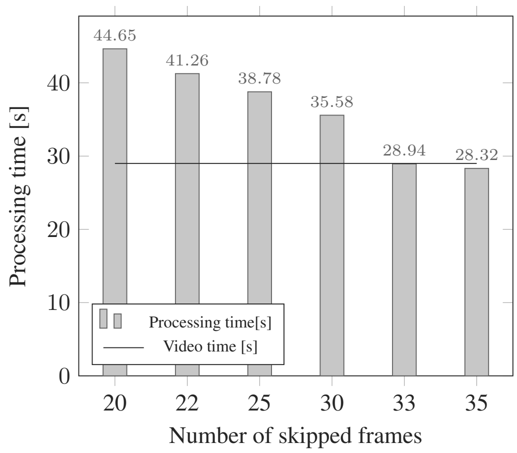


Fig. 10. Bee identification execution time when using the KNN adaptive algorithm with different numbers of skipped frames at medium (960 × 540 px) video resolution.

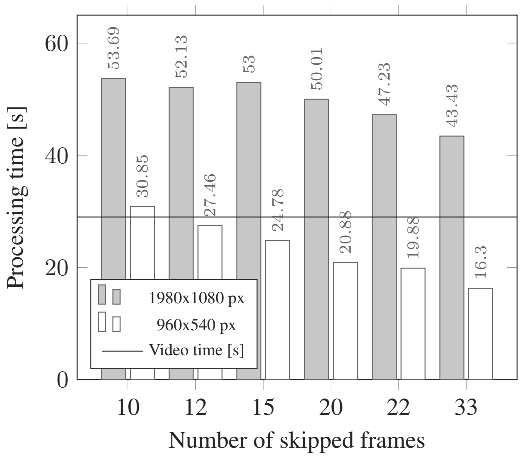


Fig. 11. Bee identification execution time when using the DetectNet CNN algorithm with different numbers of skipped frames at medium (960 × 540 px) and high (1980 × 1080 px) video resolution.

classified correctly. These results are imperfect but satisfactory. We are primarily concerned with the FN rate: accurate detection of a single case of varroosis is sufficient; FP rates are of less concern.

4.2.2. Varroosis detection time performance. We analyzed time performance of the varroosis detection process in isolation from the bee identification process. We processed a total of 907 images of different resolutions. For each image, we noted only the time required for classification by the CNN. We did not measure the time required to load each image into memory, given that during real operation the bee images passed to the classifier by the identification model would already exist in common memory. The most important performance parameter is response time—this determines whether the device can process the video stream in real time. The total detection time for the 907 images exceeded 16 s. The collected results show an average processing time of 18 ms with a standard deviation of 3 ms, independent of image size. When running both bee identification and *V. destructor* detection synchronously, the image processing will take place in real time only if the sum of the processing time of both phases is less than or equal to the duration of the video stream. Given that the bee identification run time exceeds 30 s, for a medium resolution video with 10 frames skipped, and the *V. destructor* detection exceeds 16 s, we pipelined both processes. Such an approach allowed us to maximize the usage of computing units, and ensured that both phases could be completed in real time.

The combination of *V. destructor* detection and bee identification resulted in a slightly increased processing time, when compared with bee identification alone. When skipping 11 video frames, the difference in processing time was the largest, at over 5 s. When skipping 28 frames, the difference was slightly more than 1 s. This is to be expected, given that the number of identified bees increases with the number of processed video frames. Figure 12 shows the dependence of total video processing time on the number of skipped frames.

5. Discussion

The solution presented in this paper can help beekeepers to track *V. destructor* mites within apiaries. Although this is not the first approach to precision beekeeping, existing works lack the comprehensive solution that we present: with IoT monitoring devices, edge-based *V. destructor* detection, and data transmission to a cloud data center for exhaustive monitoring and notification. As such, our approach represents a novel solution.

Marstaller *et al.* (2019) and König (2019) focus on analyzing bee images. Both groups of authors entertain the possibility of varroosis detection but do not

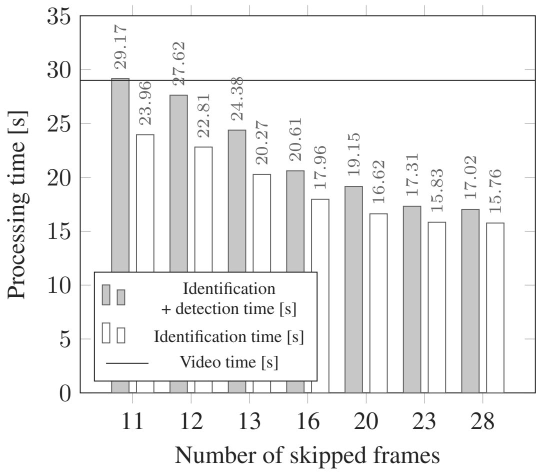


Fig. 12. Dependence between the total video processing time (both bee identification and *V. destructor* detection) and the number of skipped frames.

Table 1. Confusion matrix for *V. destructor* detection.

		Actual class	
		P	N
Predicted	P	17	2
class	N	8	23

Table 2. Efficacy metrics for *V. destructor* detection.

Metric	Value
Precision	0.89
Sensitivity	0.68
Accuracy	0.80
Specificity	0.92

provide an appropriate IoT implementation. The solution proposed by Elizondo *et al*. (2013) detects invasive mites, but focuses only on algorithmic implementations at the expense of video capture. Although the authors demonstrate improved accuracy (approximately 90%) when using a background subtraction algorithm, they only detect *V. destructor* mites within honeybee cells, not on the bodies of bees. Our proposal was designed and tested on real bees in their natural environment. Moreover, we implemented a complex system that includes image acquisition, bee identification, and machine learning bee health analysis, all performed on-edge. Our solution is effective, low-power, and non-invasive.

Furthermore, our solution produces satisfactory results when compared with alternate IoT solutions for the detection of *V. destructor* mites which do not focus on image analysis. Szczurek *et al*. (2020) applied an SVM classifier to detect *V. destructor* mites based on beehive air analysis. The authors demonstrated a sensitivity of 0.67–0.75, depending on the category of *V. destructor* infestation level (low, medium, or high), while our solution reached 0.68. Both solutions demonstrate comparable specificities of more than 0.90. Both also exhibit comparable detection efficacy, although they analyze different data types. However, with use of Nvidia Jetson Nano, our solution implements edge-based processing and analysis, thereby limiting the number of required data transmissions to the data center.

A comparison of the adaptive algorithms and the machine learning approach led to two conclusions. First, CNNs provide superior bee detection to adaptive algorithms, reflected in higher values of the classification metrics and correct detection of bee objects within video frames. Second, the CNN approach requires less processing time, leading to improved performance and the ability to analyze more video frames within a given time period. This increases the detection speed of *V. destructor* mites, as the bee can be inspected from multiple angles. Finally, we achieved satisfactory results regarding the quality of bee identification and *V. destructor* detection when using a video stream with medium resolution. Consequently, the video processing time is reduced, and our system can be used for real-time beehive monitoring.

6. Conclusion

Early detection of *V. destructor* mites within beehives plays a crucial role in beekeeping and, consequently, in food production. This paper demonstrated that such detection can be accomplished by visual inspection of bees entering a beehive, followed by image processing and analysis. We also showed that these processes could be performed in their entirety on IoT monitoring devices. Our solution used the Nvidia Jetson Nano accelerator. This enabled us to perform edge based machine learning data analysis, without the need to transmit video streams to a data analysis center, thereby improving device operating time. Although we used a high-resolution Sony IMX219-77 camera to capture video, we demonstrated experimentally that *V. destructor* mites can be detected at lower resolutions. We used a cloud data center to enable the connection of many devices, and therefore the monitoring of many beehives and apiaries. Such large-scale monitoring will hopefully provide a broader picture of *V. destructor* infestations globally.

Acknowledgment

This work was supported by pro-quality grants for highly scored publications or issued patents (02/090/RGJ21/0011, 02/100/RGJ21/0009), a professorship grant (02/020/RGP19/0184) of the Rector of the Silesian University of Technology, Gliwice, Poland, partially by statutory research funds of the Department of Applied Informatics (grant 02/100/BK_22/0017), the Department of Distributed Systems and Informatic Devices (BK-243/RAu8/2022) and the Department of Graphics, Computer Vision, and Digital Systems (BK-241/RAu6/2022), Silesian University of Technology, Gliwice, Poland, and, in part, by the Polish Ministry of Science and Higher Education within the CyPhiS program of the Silesian University of Technology, Gliwice, Poland (contract POWR.03.02.00-00-I007/17-00).

References

Babic, Z., Pilipovic, R., Risojevic, V. and Mirjanic, G. (2016). Pollen bearing honey bee detection in hive entrance video recorded by remote embedded system for pollination monitoring, *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* **III–7**: 51–57.

Balta, A., Dogan, S., Ozmen Koca, G. and Akbal, E. (2017). Software modeling of remote controlled beehive design, *International Conference on Advances and Innovations in Engineering (ICAIE), Elziğ, Turkey*, pp. 1–6.

Banks, A., Briggs, E., Borgendale, K. and Gupta, R. (2019). *MQTT Version 5.0*, OASIS Standard, https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html.

Barron, A.B. (2015). Death of the bee hive: Understanding the failure of an insect society, *Current Opinion in Insect Science* **10**: 45–50.

Bayir, R. and Albayrak, A. (2016). The monitoring of nectar flow period of honey bees using wireless sensor networks, *International Journal of Distributed Sensor Networks* **12**(11): 1–8.

Bencsik, M., Bencsik, J., Baxter, M., Lucian, A., Romieu, J. and Millet, M. (2011). Identification of the honey bee swarming process by analysing the time course of hive vibrations, *Computers and Electronics in Agriculture* **76**(1): 44–50.

Bjerge, K., Frigaard, C.E., Mikkelsen, P.H., Nielsen, T.H., Misbih, M. and Kryger, P. (2019). A computer vision system to monitor the infestation level of *Varroa destructor* in a honeybee colony, *Computers and Electronics in Agriculture* **164**: 104898.

Boecking, O. and Genersch, E. (2008). Varroosis—The ongoing crisis in bee keeping, *Journal für Verbraucherschutz und Lebensmittelsicherheit* **3**: 221–228.

Bojanic Rasovic, M., Davidović, V. and Joksimović-Todorović, M. (2018). Measures to protect bee health against varroosis in Montenegro, *Acta Agriculturae Serbica* **23**(46): 177–185.

Braga, A.R., Hassler, E.E., Gomes, D.G., Freitas, B.M. and Cazier, J.A. (2019). IoT for development: Building a classification algorithm to help beekeepers detect honeybee health problems early, *Americas Conference on Information Systems (AMCIS), Cancún, Mexico*, pp. 1–10.

Campbell, J., Mummert, L. and Sukthankar, R. (2008). Video monitoring of honey bee colonies at the hive entrance, *Workshop on Visual Observation and Analysis of Vertebrate and Insect Behaviour (ICPR), Tampa, USA,* Vol. 8, pp. 1–4.

Chen, C., Yang, E.-C., Jiang, J.-A. and Lin, T.-T. (2012). An imaging system for monitoring the in-and-out activity of honey bees, *Computers and Electronics in Agriculture* **89**: 100 – 109.

Chen, Y.-L., Chien, H.-Y., Hsu, T.-H., Jing, Y.-J., Lin, C.-Y. and Lin, Y.-C. (2020). A PI-based beehive IoT system design, *in* C.-N. Yang *et al.* (Eds), *Security with Intelligent Computing and Big-Data Services*, Springer International Publishing, Cham, pp. 535–543.

Chen, Y.P. and Siede, R. (2007). Honey bee viruses, *in* K. Maramorosch *et al.* (Eds), *Advances in Virus Research*, Academic Press, Cambridge, pp. 33–80.

Cornman, R.S., Tarpy, D.R., Chen, Y., Jeffreys, L., Lopez, D., Pettis, J.S., vanEngelsdorp, D. and Evans, J.D. (2012). Pathogen webs in collapsing honey bee colonies, *PLOS ONE* **7**(8): 1–15.

Dasig, D.D. and Mendez, J.M. (2020). An IoT and wireless sensor network-based technology for a low-cost precision apiculture, *in* P. Pattnaik *et al.* (Eds), *Internet of Things and Analytics for Agriculture*, Springer, Singapore, Vol. 2, pp. 67–92.

Debauche, O., Moulat, M.E., Mahmoudi, S., Boukraa, S., Manneback, P. and Lebeau, F. (2018). Web monitoring of bee health for researchers and beekeepers based on the Internet of things, *Procedia Computer Science* **130**: 991–998.

Dineva, K. and Atanasova, T. (2018). OSEMN process for working over data acquired by IoT devices mounted in beehives, *Current Trends in Natural Sciences* **7**(13): 47–53.

Domański, A., Domańska, J., Czachórski, T., Klamka, J., Szyguła, J. and Marek, D. (2021). The IoT gateway with active queue management, *International Journal of Applied Mathematics and Computer Science* **31**(1): 165–178, Doi: 10.34768/amcs-2021-0012.

Dunham, W. (1931). Hive temperatures for each hour of a day, *The Ohio Journal of Science* **31**(107): 181–188.

Edwards-Murphy, F., Magno, M., Whelan, P.M., O'Halloran, J. and Popovici, E.M. (2016). b+WSN: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring, *Computers and Electronics in Agriculture* **124**: 211–219.

Elizondo, V., Briceno, J., Travieso, C. and Alonso, J. (2013). Video monitoring of a mite in honeybee cells, *Advanced Materials Research* **664**: 1107–1113.

Fitzgerald, D.W., Murphy, F.E., Wright, W.M.D., Whelan, M. and Popovici, E.M. (2015). Design and development of a smart weighing scale for beehive monitoring, *2015 26th Irish Signals and Systems Conference (ISSC), Carlow, Ireland*, pp. 1–6.

Gates, B.N. (1914). *The Temperature of the Bee Colony*, US Department of Agriculture, Washington DC.

Gil-Lebrero, S., Quiles Latorre, F., Ortiz, M., Sánche Ruiz, V., Gamiz, V. and Luna-Rodriguez, J.-J. (2017). Honey bee colonies remote monitoring system, *Sensors* **17**(1): 55.

Gołosz, M. and Mrozek, D. (2019). Exploration of data from smart bands in the cloud and on the edge—The impact on the data storage space, *in* J.M.F. Rodrigues *et al.* (Eds), *Computational Science—ICCS 2019*, Springer International Publishing, Cham, pp. 607–620.

Grzesik, P. and Mrozek, D. (2021). Metagenomic analysis at the edge with Jetson Xavier NX, *in* M. Paszynski *et al.* (Eds), *Computational Science—ICCS 2021*, Springer International Publishing, Cham, pp. 500–511.

Guzmán-Novoa, E., Eccles, L., Calvete, Y., Mcgowan, J., Kelly, P.G. and Correa-Benítez, A. (2010). *Varroa destructor* is the main culprit for the death and reduced populations of overwintered honey bee *(Apis mellifera)* colonies in Ontario, Canada, *Apidologie* **41**(4): 443–450.

König, A. (2019). Indusbee 4.0—Integrated intelligent sensory systems for advanced bee hive instrumentation and hive keepers' assistance systems, *Sensors & Transducers Journal* **237**(9–10): 109–121.

Kontogiannis, S. (2019). An Internet of things-based low-power integrated beekeeping safety and conditions monitoring system, *Inventions* **4**(3): 1–26.

Krzyśka, R. (2022). ULmonitor, `http://ulmonitor.pl/onas-eng.htm`.

Kviesis, A. and Zacepins, A. (2015). System architectures for real-time bee colony temperature monitoring, *ICTE in Regional Development, Valmiera, Latvia,* pp. 86–94.

Machhamer, R., Altenhofer, J., Ueding, K., Czenkusch, L., Stolz, F., Harth, M., Mattern, M., Latif, A., Haab, S., Herrmann, J., Schmeink, A., Gollmer, K. and Dartmann, G. (2020). Visual programmed IoT beehive monitoring for decision aid by machine learning based anomaly detection, *2020 9th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro*, pp. 1–5.

Marstaller, J., Tausch, F. and Stock, S. (2019). DeepBees—Building and scaling convolutional neuronal nets for fast and large-scale visual monitoring of bee hives, *International Conference on Computer Vision, Seoul, Republic of Korea*, pp. 1–8.

Meikle, W. and Holst, N. (2014). Application of continuous monitoring of honeybee colonies, *Apidologie* **46**(1): 10–22.

Meitalovs, J., Histjajevs, A. and Stalidzans, E. (2009). Automatic microclimate controlled beehive observation system, *8th International Scientific Conference 'Enginieering for Rural Development', Jelgava, Latvia,* pp. 265–271.

Mielnik, P., Fojcik, M., Tokarz, K., Rodak, Z. and Pollen, B. (2021). Detecting of minimal changes in physical activity using one accelerometer sensor, *in* K. Wojtkiewicz *et al.* (Eds), *Advances in Computational Collective Intelligence*, Springer International Publishing, Cham, pp. 498–508.

Mielnik, P., Tokarz, K., Mrozek, D., Czekalski, P., Fojcik, M., Hjelle, A.M. and Milik, M. (2019). Monitoring of chronic arthritis patients with wearables—Report from the concept phase, *in* N.T. Nguyen *et al.* (Eds), *Computational Collective Intelligence*, Springer International Publishing, Cham, pp. 229–238.

Mishra, B. and Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: A survey, *IEEE Access* **8**: 201071–201086.

Mrozek, D., Milik, M., Małysiak-Mrozek, B., Tokarz, K., Duszenko, A. and Kozielski, S. (2020a). Fuzzy intelligence in monitoring older adults with wearables, *in* V.V. Krzhizhanovskaya *et al.* (Eds), *Computational Science—ICCS 2020*, Springer International Publishing, Cham, pp. 288–301.

Mrozek, D., Tokarz, K., Pankowski, D. and Małysiak-Mrozek, B. (2020b). A hopping umbrella for fuzzy joining data streams from IoT devices in the cloud and on the edge, *IEEE Transactions on Fuzzy Systems* **28**(5): 916–928.

Nazir, D., Fizza, M., Waseem, A. and Khan, S. (2018). Vehicle detection on embedded single board computers, *2018 7th International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia*, pp. 480–485.

Nvidia (32.7.1 Release). *Nvidia Jetson Linux Developer Guide*, Nvidia, Santa Clara, `https://docs.nvidia.com/jetson/l4t/`.

Ochoa, I.Z., Gutierrez, S. and Rodríguez, F. (2019). Internet of things: Low cost monitoring beehive system using wireless sensor network, *2019 IEEE International Conference on Engineering Veracruz (ICEV), Boca del Rio, Mexico*, Vol. I, pp. 1–7.

Pierleoni, P., Concetti, R., Belli, A. and Palma, L. (2020). Amazon, Google and Microsoft solutions for IoT: Architectures and a performance comparison, *IEEE Access* **8**: 5455–5470.

Qandour, A., Ahmad, I., Habibi, D. and Leppard, M. (2014). Remote beehive monitoring using acoustic signals, *Acoustics Australia/Australian Acoustical Society* **42**(3): 204–209.

Rodak, Z., Tokarz, K., Mielnik, P. and Fojcik, M. (2022). Simultaneous measurements reading from more than one Mi Band 3 wristbands, *in* A.K. Nagar *et al.* (Eds), *Intelligent Sustainable Systems*, Springer Singapore, Singapore, pp. 93–101.

Rustia, D.J., Ngo, N. and Lin, T.-T. (2016). An IoT-based information system for honeybee in and out activity with beehive environmental condition monitoring, *Conference on Bio-Mechatronics and Agricultural Machinery Engineering, Niigata, Japan*, pp. 1–2.

Schneider, P. and Drescher, W. (1987). Einfluss der parasitierung durch die milbe varroa jacobsoni oud. auf das schlupfgewicht, die gewichtsentwicklung, die entwicklung der hypopharynxdrüsen und die lebensdauer von Apis mellifera l, *Apidologie* **18**(1): 101–110.

Schurischuster, S., Remeseiro, B., Radeva, P. and Kampel, M. (2018). A preliminary study of image analysis for parasite detection on honey bees, *in* A. Campilho *et al.* (Eds), *Image Analysis and Recognition. ICIAR 2018*, Lecture Notes in Computer Science, Vol. 10882, Springer, Cham, pp. 465–473.

Schurischuster, S., Zambanini, S. and Kampel, M. (2016). Sensor study for monitoring varroa mites on honey bees (*Apis mellifera*), *Visual Observation and Analysis of Vertebrate and Insect Behavior Workshop, Cancun, Mexico,* pp. 1–4.

Stefanowski, J., Krawiec, K. and Wrembel, R. (2017). Exploring complex and big data, *International Journal of Applied Mathematics and Computer Science* **27**(4): 669–679, DOI: 10.1515/amcs-2017-0046.

Szczurek, A., Maciejewska, M., Wilk, J., Wilde, J. and Siuda, M. (2019). Detection of honeybee disease: Varrosis using a semiconductor gas sensor array, *8th International Conference on Sensor Networks, Prague, Czech Republic*, pp. 58–66.

Szczurek, A., Maciejewska, M., Zajiczek, Z., Wilk, J., Wilde, J. and Siuda, M. (2020). The effectiveness of *Varroa destructor* infestation classification using an e-nose depending on the time of day, *Sensors* **20**(9): 2532.

Süzen, A.A., Duman, B. and Şen, B. (2020). Benchmark analysis of jetson TX2, Jetson Nano and Raspberry Pi using deep-CNN, *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey*, pp. 1–5.

van der Sluijs, J.P., Simon-Delso, N., Goulson, D., Maxim, L., Bonmatin, J.-M. and Belzunces, L.P. (2013). Neonicotinoids, bee disorders and the sustainability of pollinator services, *Current Opinion in Environmental Sustainability* **5**(3): 293–305.

Van Goethem, S., Verwulgen, S., Goethijn, F. and Steckel, J. (2019). An IoT solution for measuring bee pollination efficacy, *2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland*, pp. 837–841.

vanEngelsdorp, D., Traynor, K.S., Andree, M., Lichtenberg, E.M., Chen, Y., Saegerman, C. and Cox-Foster, D.L. (2017). Colony collapse disorder (CCD) and bee age impact honey bee pathophysiology, *PLOS ONE* **12**(7): 1–23.

vanEngelsdorp, D., Underwood, R., Caron, D. and Hayes, J. (2007). An estimate of managed colony losses in the winter of 2006–2007: A report commissioned by the apiary inspectors of America, *American Bee Journal* **147**(7): 599–603.

Wojnakowski, M., Wiśniewski, R., Bazydło, G. and Popławski, M. (2021). Analysis of safeness in a Petri net-based specification of the control part of cyber-physical systems, *International Journal of Applied Mathematics and Computer Science* **31**(4): 647–657, DOI: 10.34768/amcs-2021-0045.

Zabasta, A., Zhiravetska, A., Kunicina, N. and Kondratjevs, K. (2019). Technical implementation of IoT concept for bee colony monitoring, *2019 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro*, pp. 1–4.

Zacepins, A., Kviesis, A., Pecka, A. and Osadcuks, V. (2017a). Development of Internet of things concept for precision beekeeping, *2017 18th International Carpathian Control Conference (ICCC), Sinaia, Romania*, pp. 23–27.

Zacepins, A., Pecka, A., Osadcuks, V., Kviesis, A. and Engel, S. (2017b). Solution for automated bee colony weight monitoring, *Agronomy Research* **15**(2): 585–593.

Zgank, A. (2020). Bee swarm activity acoustic classification for an IoT-based farm service, *Sensors* **20**(1): 1–14.

Zgank, A. (2021). IoT-based bee swarm activity acoustic classification using deep neural networks, *Sensors* **21**(3): 1–14.

Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction, *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK,* Vol. 2, pp. 28–31.

Zivkovic, Z. and van der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction, *Pattern Recognition Letters* **27**(7): 773–780.

Zogovic, N., Mladenovic, M. and Rašić, S. (2017). From primitive to cyber-physical beekeeping, *7th International Conference on Information Society and Technology, Kopaonik, Serbia*, Vol. 1, pp. 38–43.

Anna Wachowicz received her MSc degree in computer science from the Silesian University of Technology in Gliwice, Poland, in 2018. She is a PhD student within the Cyber-Physical Studies program at the Silesian University of Technology. She also works for the Vattenfall company as a software developer. Her interests cover algorithmic trading, big data and agile software development.

Jakub Pytlik received his MSc degree in computer science from the Silesian University of Technology in Gliwice, Poland, in 2020. He currently works for the Pitney Bowes company as a senior software engineer. His interests cover .NET programming, microservices-based software and machine learning.

Bożena Małysiak-Mrozek received her MSc and PhD degrees in computer science from the Silesian University of Technology in Gliwice, Poland. She is an associate professor at the Department of Distributed Systems and Informatic Devices at the Silesian University of Technology. Her scientific interests cover the IoT, information systems, computational intelligence, bioinformatics, databases, big data, cloud computing, and soft computing methods. She has been a co-editor of 15 books devoted to databases and data processing.

Krzysztof Tokarz holds MSc and PhD degrees in computer science from the Silesian University of Technology in Gliwice, Poland. He is an assistant professor in the Institute of Informatics in the Faculty of Automatic Control, Electronics and Computer Science at the Silesian University of Technology. His scientific interests cover microprocessor and embedded systems, low-level programming languages, wireless computer networks, and the Internet of things.

Dariusz Mrozek is currently an associate professor and the head of the Department of Applied Informatics at the Silesian University of Technology in Gliwice, Poland. His research interests cover the IoT, parallel and cloud computing, databases, big data, and bioinformatics. He is the author of 150+ papers published in international journals and conference proceedings, the author of two books on the use of big data analytics and high-performance computations in protein bioinformatics published by Springer, a co-editor of 15 other books devoted to databases and data processing, and the editor of many special issues in reputable scientific journals. He has collaborated with esteemed institutions by working on different research projects, including the Imperial College of London, Microsoft Research and Amazon in the USA.

Received: 30 September 2021
Revised: 29 April 2022
Re-revised: 19 May 2022
Accepted: 20 May 2022