electronics

MDPI

Article

Integrating IoT and Image Processing for Crop Monitoring: A LoRa-Based Solution for Citrus Pest Detection

Joel L. Quispe-Vilca [1], Edison Moreno-Cardenas [2,*], Erwin J. Sacoto-Cabrera [3] and Yackelin Moreno-Cardenas [4]

[1] Networks and Computers Research Group (GRC), Universitat Politècnica de València, 46022 Valencia, Spain; joequivi@posgrado.upv.es
[2] Communications Department, Universitat Politècnica de València, 46022 Valencia, Spain
[3] GIHP4C, Universidad Politécnica Salesiana, Cuenca 010105, Ecuador; esacoto@ups.edu.ec
[4] Departamento de Ingeniería Ambiental, Universidad Nacional Jose María Arguedas, Andahuaylas 03701, Peru; ymmoreno@unajma.edu.pe
* Correspondence: edmocar@doctor.upv.es

Abstract: Today, agriculture faces many challenges, such as the use of inefficient methods that affect crop quality. Precision agriculture (PA), combined with advanced technologies, improves monitoring, while the integration of wireless communication optimizes processes and resources. This work presents the design of a communication prototype applied in precision agriculture, which allows the acquisition, processing, and wireless transmission of information extracted from the Cotonet pest to The Things Network (TTN) cloud server. This prototype integrates technologies and protocols such as LoRaWAN, Message Queuing Telemetry Transport (MQTT), Internet of Things (IoT) sensors, and Computer Vision. This prototype employs a robust processing and segmentation algorithm, which allows the recognition of pests in citrus plants based on color. The results show that lighting conditions, weather, and time of day influence the quality of the captured images. The relationship between image resolution, brightness, and processing time shows that higher-resolution images (1920 × 1080 pixels per image) provide better detection of pest pixels (greater than 50% of the pest index) but require longer processing time (28.415 ms on average). Furthermore, the developed system effectively detects an index of affection of *Planococcus citri* (Cotonet) in agricultural plantations through an end-to-end technological implementation that integrates image processing, wireless communication, and IoT technologies.

Keywords: LoRa; LoRaWAN; image processing; precision agriculture; communication network

check for updates

Citation: Quispe-Vilca, J.L.; Moreno-Cardenas, E.; Sacoto-Cabrera, E.J.; Moreno-Cardenas, Y. Integrating IoT and Image Processing for Crop Monitoring: A LoRa-Based Solution for Citrus Pest Detection. *Electronics* **2024**, *13*, 4863. https://doi.org/10.3390/electronics13244863

Academic Editor: Antoni Morell

Received: 31 October 2024
Revised: 29 November 2024
Accepted: 2 December 2024
Published: 10 December 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, the inclusion of modern technologies in agriculture, particularly precision agriculture (PA), has revolutionized crop monitoring and management, as described [1,2]. Although traditional farming methods have historically been influential, they often fall short in terms of resource efficiency, pest control, crop monitoring, and product quality [3]. These constraints have driven the adoption of cutting-edge technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and Data Analytics, advancing the Digital Transformation (DT) of agriculture [4]. These advances have the potential to significantly boost agricultural productivity and sustainability, heralding a promising future for the industry.

Pest control remains a crucial issue in agriculture, as pest invasions cause significant damage to crops and reduce farm profitability, as described in [5]. Effective pest management through the use of technology is essential to maintain crop health and optimize pesticide use, helping to minimize environmental damage while ensuring consumer safety, as described in [4]. Traditional pest detection methods, such as manual inspections, are time-consuming and labor-intensive. Sial et al. [6] described that these methods often do not

allow for timely intervention, resulting in significant crop damage before the problem can be addressed.

To overcome these challenges, IoT and fog computing have emerged as essential tools, giving farmers immediate access to data on field conditions, including pest activity [7]. The real-time data collection and processing capabilities of IoT technologies give farmers the speed and efficiency needed to make timely and informed decisions. Moreover, given that many agricultural areas are located far from data processing centers, the ability of IoT and fog computing to transmit data over long distances with low power consumption is transformative [8,9]. These technologies address power and connectivity limitations, making them ideal for large agricultural fields and paving the way for a more connected and efficient future in agriculture [10,11].

This paper presents the development and implementation of a prototype AP system based on an IoT sensor network and image processing for citrus pest monitoring. Therefore, this work's main objective is designing a communication prototype for AP, which allows the acquisition, processing, and wireless transmission of data for detecting the Cotonet pest (*Planococcus citri*) in citrus plants. The system also integrates LoRaWAN technology, which allows for the transmission of data captured by the camera sensors and their subsequent processing for pest detection. The main contributions of this work are as follows:

- Development of a robust and low-cost communication prototype for remote detection of Cotonet pests in citrus crops using LoRaWAN for long-range, energy-efficient data transmission in agricultural environments.
- Implementation of a color-based image processing technique that improves the accuracy of pest recognition by focusing on specific shades associated with Cotonet pests, improving detection reliability under variable illumination.
- Optimization of resource consumption based on performing initial image processing at the node level, reducing the need for extensive data transmission and extending device battery life and communication reliability for sustained agricultural monitoring.

In this article, we apply IoT network concepts and image processing techniques to formulate a prototype solution for the detection of citrus pests. The LoRaWAN IoT network is used in the prototype to gather data from citrus crop monitoring and transmit it to a server for storage and analysis. The LoRaWAN IoT network is a branch of the Low-Power Wide-Area Network (LPWAN) protocol, such as SigFox and NB-IoT, enabling long-range data transmission with minimal energy use [12]. However, LoRaWAN versus NB-IoT and Sigfox prove to be suitable for remote farming environments, and while NB-IoT supports higher data rates and larger payloads, it has a higher cost due to their licensed spectrum, while Sigfox is more restrictive concerning data rate and payload size. LoRaWAN operating in unlicensed frequency bands is ideal for IoT applications such as smart cities [13,14], smart agriculture [15,16], and industrial monitoring [17,18], offering extended battery life for connected devices [17]. Moreover, in this article, color image processing techniques are utilized to detect citrus pests. Specifically, digital image processing utilizes computer algorithms to analyze, enhance, or manipulate images through a range of techniques, addressing issues such as noise and signal distortion that occur during the processing stages, as highlighted in [19] . Additionally, our work addresses key challenges and unresolved issues in crop monitoring research [20,21], positioning it within a contemporary research trend. This aligns with several of the studies discussed in the following subsection:

Related Work

Many works represent PA as a paradigm shift in modern agricultural practices, focusing on optimizing resource management and improving crop yields through advanced technologies [2]. In this sense, the literature highlights several PA-focused studies using IoT and image processing technologies for crop monitoring [22,23]. Several studies explore the impact of technologies in various scenarios aimed at improving both agricultural productivity and overall farm management. For instance, in the context of PA, in [24],

information technologies (ITs) are crucial for effective decision making by integrating technical, economic, and environmental management perspectives. These technologies, which rely on site-specific data collected from sensors that transmit real-time information through communication networks, enable farmers to apply management practices tailored to the unique requirements of each field [25]. The study by the authors in [26] examines the impact of implementing PA techniques in Central Europe, revealing a 15% increase in crop yields as a result of these advancements. In [27], the authors explore the fundamental goal of PA, which is to increase the efficiency of agricultural inputs such as water, fertilizers, and pesticides while minimizing the environmental impact. This approach is enhanced by the application of machine learning techniques [28].

In the field of sensors and IoT networks, several studies have described IoT implementations in various agricultural scenarios designed specifically for farmers. The critical role of PA in real-time monitoring of physical and chemical soil parameters such as moisture, pH, and temperature through wireless sensor networks is analyzed in [29]. Specifically, digital image processing utilizes computer algorithms to analyze, enhance, or manipulate images through a range of techniques, addressing issues such as noise and signal distortion that occur during the processing stages, as highlighted in [19]. Specifically, the authors in [2] indicate that LoRaWAN has enabled the transformation of precision agriculture by providing long-range, low-power communication that covers vast agricultural areas in a way. Similarly, the authors in [30] present an evaluation of LoRaWAN's performance in tropical agricultural environments, highlighting the impact of dense vegetation on signal propagation, including range, spreading factor, and coding rate. The findings indicate that while LoRa supports real-time agricultural monitoring, its performance is significantly influenced by obstacles and environmental conditions. Additionally, Singh et al. [31] implemented a LoRaWAN-based system for monitoring greenhouses, emphasizing precision agriculture. Their contributions include an airflow-optimized sensor box for improved accuracy, a low-power wireless network for continuous crop monitoring, and a customized dashboard for data visualization and decision making. This work highlights the effective integration of IoT technologies to optimize agricultural management in controlled environments.

LoRaWAN ensures low data rates, which reduces power consumption, making it ideal for field sensors that rely on minimal battery usage, as described in [32,33]. The LoRaWAN architecture also supports bidirectional communication and secure data transmission, as detailed in [34]. In this context, the authors of [34] present an innovative system that integrates UAVs with technologies such as LoRa and ZigBee to address challenges related to coverage and energy consumption. The experimental results validate the system's effectiveness in both static and dynamic operations, while evolutionary algorithms enhance UAV trajectory planning to achieve optimal efficiency. Finally, LoRaWAN, as a cloud-based protocol, efficiently manages connections across regional, national, or global networks, addressing key IoT needs such as bidirectional communication, mobility, and data security [35]. Its architecture enables long-distance data transmission while optimizing network parameters like data rate and transmission power. This ensures continuous, reliable data collection, ultimately enhancing resource management and crop monitoring, as described in [36].

Within the field of image processing, it has been gaining ground by integrating advanced technologies such as machine vision, deep learning, and artificial intelligence [37]. Several studies have explored the use of remote sensing, drones, and IoT-based sensors for real-time data acquisition and crop monitoring, as described in [22]. For example, in [38], a system is presented that combines Wireless Underground Sensor Networks (WUSNs) and UAVs to collect data in areas without network coverage. Using LoRa for efficient underground communication and ZigBee for data transfer, experiments validate its feasibility, while algorithms optimize UAV trajectories for efficiency. In the same sense, in [39], the authors describe how image processing in PA, combined with AI and machine learning, is used to monitor crop growth, detect diseases, and estimate yields. High-resolution images are captured by drones or IoT platforms and then analyzed to provide actionable insights. These insights help optimize irrigation, nutrient delivery, and overall crop management.

In [40], the authors discuss the application of image processing techniques in agriculture to analyze plant cover, biomass, and leaf area index, aiming to enhance irrigation practices, nutrient application, and overall crop management. Finally, in [41], the authors use an image processing and machine learning approach with the objective of reviewing and examining various IoT-based image processing and machine learning techniques for the monitoring of agricultural cocoa crops.

The studies mentioned above describe the use of IoT and image processing in various PA applications separately. In contrast, our article presents the development of an end-to-end integration prototype, from sensors to the implementation of the LoRaWAN network and image processing for citrus pest detection. In addition, the prototype employs a robust algorithm for processing and segmentation, allowing for the recognition of pests on citrus plants based on color.

The remainder of this article is organized as outlined in the following paragraph, and a summary of the related work is provided below. In Section 2.1, we describe the system model proposed and its analysis, which includes the details of the brightness conditions, the image detection and processing process, The Things Network (TTN) infrastructure, and an overview of the LoRaWAN communication processes. In Section 3, we present and discuss the results obtained. In Section 4, we present this research study's conclusions. Finally, in Section 5, we suggest future directions.

2. Materials and Methods

2.1. System Model

We propose a system model based on Long-Range (LoRa) technology and computer vision with low cost, low power consumption, and long-range capabilities for detecting an affection index of *Planococcus citri* (Cotonet) in agricultural plantations. The network infrastructure of the proposed system model is based on several components, such as an image sensor, a computer that performs the image processing, a microcontroller for communication and LoRaWAN connection, a gateway to link the LoRa networks and IoT, and a connection to a TTN cloud storage server based on IoT devices to which the end user can connect, so that the end user can review the data log sent by the proposed prototype. Figure 1 shows the network architecture of the system model, as well as the distribution of the network components and the technologies used in certain parts of the architecture. In addition, one of the applications of LoRaWAN includes the use of IoT for remote environmental monitoring [42]. The image registration of the system model is based on algorithms that allow a level of efficiency and fidelity depending on the application.

The system model is comprehensively described in four subsubsections. The first subsubsection details the brightness conditions; the second focuses on the image detection and processing process, and the third explains the TTN network infrastructure, its configuration, and the location and interaction of the different devices associated with transmission over the LoRaWAN network. Finally, the fourth subsubsection provides a detailed overview of the LoRaWAN communication processes.

On the other hand, the Turia Garden, in the city of Valencia, Spain, serves as a working space for image detection. This location has the specific characteristics necessary to recognize the object of interest (pest). It is an environment of citrus trees in which a high level of heat and humidity can be appreciated and in which the pest is present. This pest has a white color and differs from the other elements of the trees (see Figure 2).

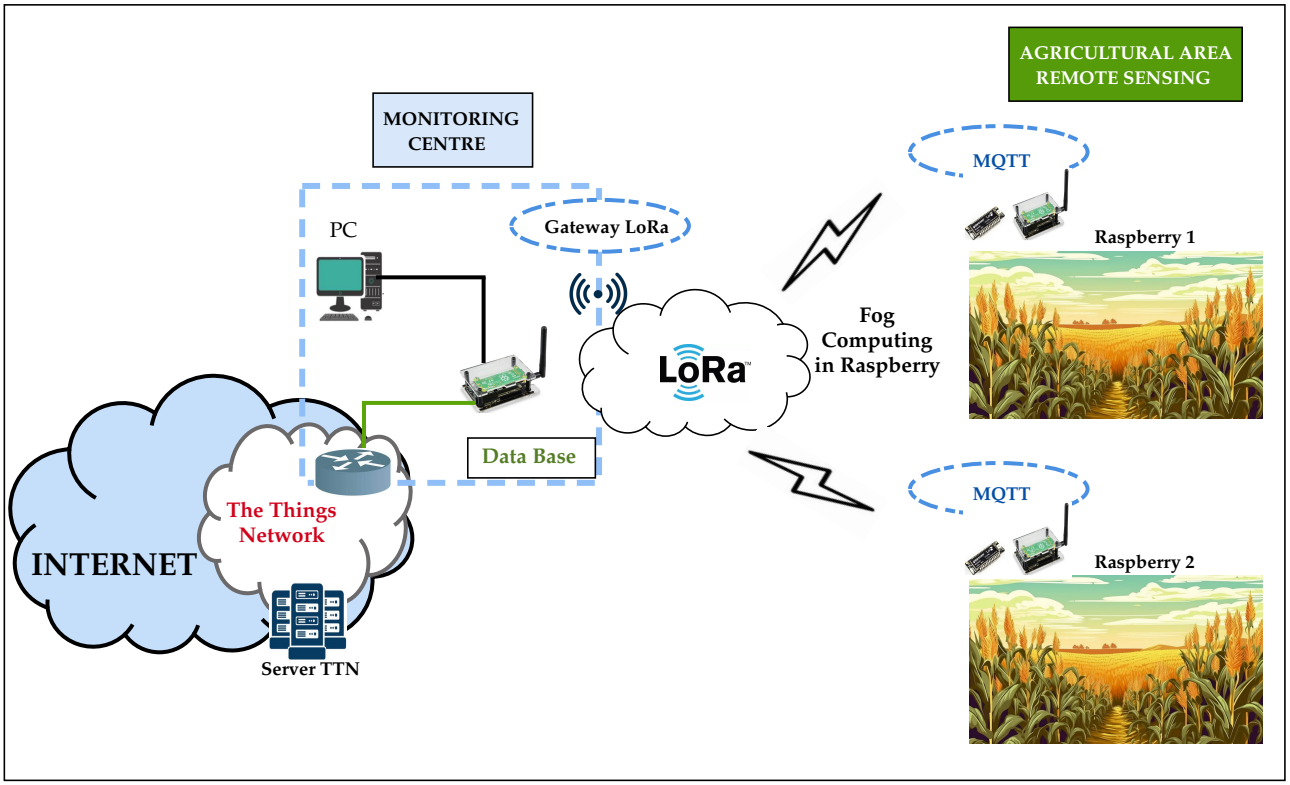


Figure 1. The network architecture of the system model.



(**a**) (**b**)

Figure 2. *Planococcus citri* (Cotonet). (**a**) The adult Cotonet pest. This image was captured with a digital camera as a microscopic-scale view. (**b**) Panoramic view of the experimental tree, where a concentration of the white Cotonet pest can be seen in the center of the image, specifically on the tree's stems. This pest represents our object of detection.

2.1.1. Brightness Conditioning

This subsubsection details the brightness conditioning of our working scenario.

The influence of lighting on image capture presents several challenges [43], from prolonged bursts to noise, which is observed in the working scenario, where lighting conditions are highly variable. In addition, weather, wind, temperature, and the presence of animals influence the efficiency and limitations of the image recording and processing system. For instance, Figure 3a shows the poorly captured image due to low illumination, and Figure 3b shows the poorly captured image due to high illumination during its capture. In both cases, the presence of winds and the instability of the proposed prototype are presented, so the image does not correspond to the area of interest.



(**a**) (**b**)

Figure 3. Poorly captured images. (**a**) Capture in low light. (**b**) Capture in high light.

Therefore, image collection is carried out during the summer, using a Raspberry Pi to analyze multiple samples. Images are captured at different times, i.e., morning, afternoon, and evening. However, low and excessive illumination causes physical saturation of the camera sensor, significantly reducing the captured images' quality and creating complications in the developed processing algorithms. Consequently, the algorithms used are not efficient in finding the object of interest (Cotonet pest) or mixing the object of interest with false positives. Thus, these difficulties in pest identification lead to false positives in the mixture of unwanted elements. Therefore, the solution to this problem is to identify the right conditions for capturing the images. These conditions are clear, windless days with sufficient sunlight. Therefore, image acquisition is carried out at specific times, namely between 19:00 and 21:00 h, as the position of the sun provides stable and controlled illumination. Thus, images acquired outside the 19:00 to 21:00 h interval are discarded for further analysis. This conditioning reduces adverse effects on the vision system, a key component in image capture and processing.

Figure 4 shows a test of several well-captured images under the conditions mentioned above.



Figure 4. Well-captured image within luminance conditions.

These images are saved and processed; finally, the representative value is extracted to move on to the communication part.

2.1.2. Detection Process

This subsubsection describes the pest detection process, addressing aspects such as image acquisition, image processing, segmentation, and feature extraction.

Figure 5 shows the system flow diagram. This diagram describes the operation of the code designed for image sensing and processing, and the transport of the information to the cloud. This system consists of two blocks of procedures for image processing. Both blocks focus on applying computer vision and fog computing techniques with the Raspberry Pi and applying LoRaWAN [42] technologies in the Lopy and MQTT (Message Queuing Telemetry Transport) over Wi-Fi and IP (Internet Protocol) networks for IoT communication.

The Lopy and Raspberry Pi devices are in charge of sending the information, while the MQTT protocol is an ISO standard based on a publish/subscribe mechanism [44,45].

The detection process is divided into three phases: the first is image acquisition, the second is image processing and segmentation, and the third is feature extraction. These phases are described below.

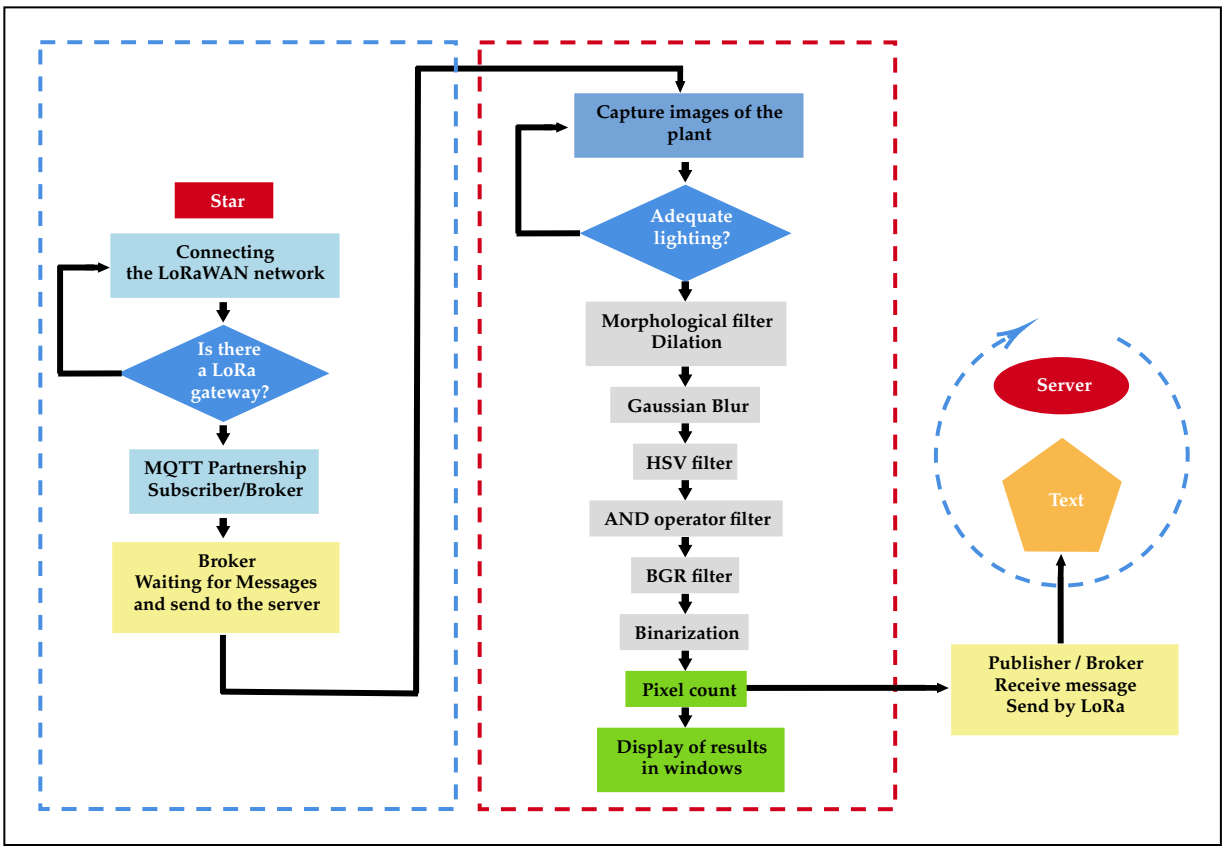


Figure 5. System flow diagram.

Phase 1: Image Acquisition

This phase describes the acquisition of images as part of the machine vision process using a Logitech C920 webcam (Logitech International S.A., Lausanne, Switzerland), with a configuration of frontal shooting conditions and short distances of 0.3 to 0.5 m. In addition, the thresholding parameters corresponding to the image processing filters within the developed algorithm are set. Likewise, the sharpness, contrast, brightness, saturation, exposure, and gain parameters are adjusted by means of the Raspberry Pi, where the video4linux device driver is installed [46]. The system works with the Python OS module, which determines the validity of the image based on its average luminance, Y, where Y is the component of luminance in Euclidean space [47,48]. Luminance analysis is necessary because excessive illumination in computer vision can saturate the Charge-Coupled Device (CCD) sensor, reducing image quality and generating false positives. Thus, an adjustment of the iris integrated into the optical system or an adjustment of the exposure time is required if the camera allows it. Therefore, the acquisition of the appropriate parameters in the digital camera is based on the strategy proposed in [49], which consists of determining whether a given image is valid for the identification of green plants (crops and weeds) based on their high-intensity levels produced by excess illumination. Therefore, the mean luminance is required, and its Y value is modeled by converting the captured Red Green Blue (RGB) image to grayscale using Equation (1).

$$Y = 0.30R + 0.59G + 0.11B \quad (1)$$

where R is red, G is green, and B is blue. The image is defined in the Y plane as a two-dimensional function $f(x)$, where $x = (x_1, x_2)$ are the spatial coordinates, and f is the luminance value of the image at any x. In addition, the image is swept, and the weighted mean is determined by $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ for n numbers, the result of which gives a value between [0, 255] due to working with 8-bit data.

Phase 2: Processing and Segmentation

This phase describes the filtering and highlighting the region of interest (ROI) by recognizing the color reflected on the tree. The size and shape part is discarded because this pest is not correlated with its expansion and deployment on the plant. In addition, image smoothing is performed to reduce noise and increase the algorithm's accuracy. Additionally, the HSV (Hue Saturation Vue) filter is applied, and then thresholding is performed, followed by the AND operator filter to eliminate the remains of pixels considered as noise, then an RGB color separation is performed so that the Open Source Computer Vision Library (OpenCV) can distinguish the pest and, finally, the image is binarized to work on the extraction. The development of the algorithm consists of six stages, as shown in Figure 6.

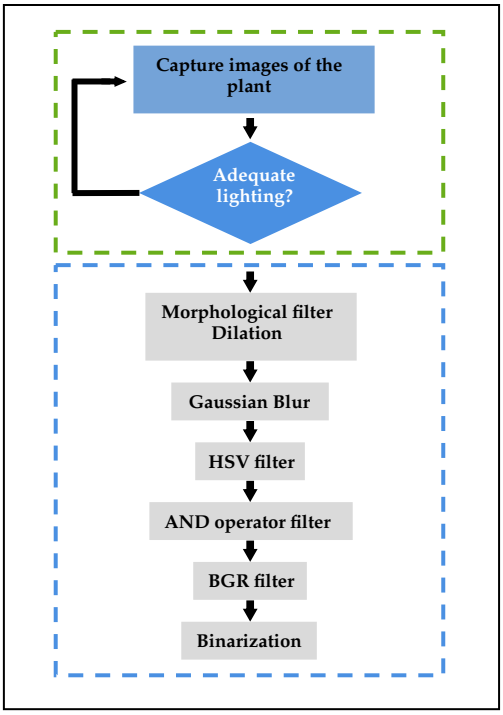


Figure 6. Processing and segmentation.

The algorithm starts with the morphological filter, which allows the best features to be highlighted using a type of mathematical morphology called dilation. This filter enables the enlargement of one pixel around the circumference of each region [50]. Furthermore, the function used is predetermined in the OpenCV library, and the dilation is given by Equations (2) and (3). These equations describe that all possible vectors are the sum of pairs of elements, one belonging to A and one to B.

$$\mathbf{A} \oplus \mathbf{B} = \{\, \mathbf{a} + \mathbf{b} \;:\; \mathbf{a} \in \mathbf{A} \quad y \quad \mathbf{b} \in \mathbf{B} \,\} \tag{2}$$

$$X \in \mathbf{A} \oplus \mathbf{B} \Leftrightarrow x = \mathbf{a} + \mathbf{b},\ \mathbf{a} \in \mathbf{A} \quad y \quad \mathbf{b} \in \mathbf{B} \tag{3}$$

where A and B are geometric structures of continuous or discrete space.

Then, it continues with the Gaussian filter, which allows smoothing of the image through the filtering process to reduce the intensity of the whites coming from the sunlight, which is considered noise, and other noises generated by the morphological filter. For this part, Gaussian blur and the linear filter defined in Equation (4) are used, where the value of an output pixel, $g(i,j)$, is determined as a weighted sum of the values of the input pixels, $f(i+k, j+l)$.

$$g(i,j) = \sum_{k,l} f(i+k, j+l)\, h(k,l) \tag{4}$$

where $h(k,l)$ is called the kernel and are the filter coefficients [51].

Then, it continues with an HSV filter [52], which allows for obtaining only the stem and the pest, eliminating unwanted objects from the plant, such as leaves and light. Based on this, the percentage of pest infestation on the stem is calculated. Default values are used for the filter, ranging from [0, 255] as 8-bit data; these values are adjusted to the values of the analyzed images with an HSV value selector paper.

Then, stem and pest segmentation is performed using the filtering method on the RGB channels. However, as the range of HSV values has already been normalized, a return function is applied to the original RGB image, subtracting the uninteresting elements previously removed with the previous filters. For this purpose, the bit-by-bit AND mathematical operator [53] is used.

Finally, color segmentation in RGB space is applied since OpenCV is applied first to the blue channel, which is not commonly used in RGB. The range of values to be filtered is completed using the Paint image viewer (version 11.2408.30.0), indicating the position of the pixels to be found; based on that position, RGB value query commands are executed in the Python framework inside the Raspberry Pi to find the set of unwanted values. From these values, a coherent range of values is selected, where minimum and maximum values are assigned to the filter values in the code, which represent the values in the RGB space of the stem; in this way, the pest is binarized (black and white) and processed as a two-dimensional array.

Phase 3: Feature Extraction

Finally, in this phase, the image is passed to a two-dimensional matrix to search for the white pixels representing the pest and calculate the crop's pest affection factor. This factor is determined by scanning the image data matrix processed in the previous phase. The white pixel count indicates that approximately 2000 white pixels are 100% of the pest infestation on the plant, and 0% are the 0 pixels found. For example, Figure 7 shows a console printout of the data representing the processed images.

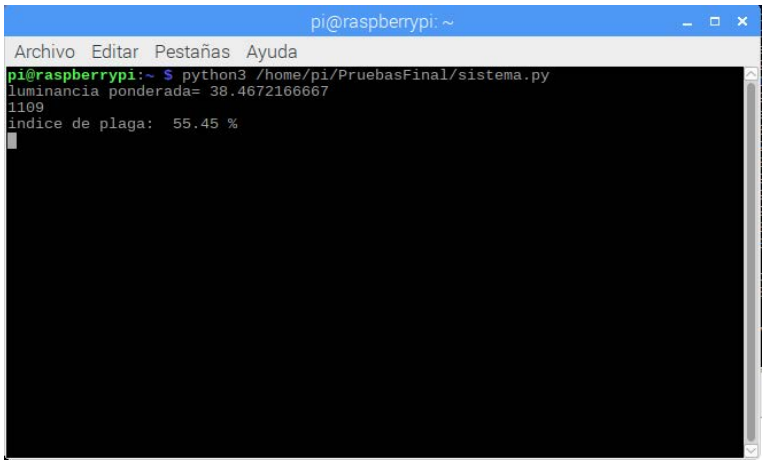


Figure 7. Representation of the image processed by computer vision.

2.1.3. TTN Infrastructure

The TTN network is the IoT platform on which this work is developed. This TTN network defines the IoT cloud platform for the LoRa protocol. It allows low-power devices to use long-range gateways to connect to a decentralized network to exchange data with applications [54]. The communication architecture in the TTN platform is shown in Figure 8. In this architecture, Raspberry Pi and Lopy devices are networked using Wi-Fi technology, which enables the MQTT layer, and both are end nodes. Meanwhile, in the LoRaWAN protocol, the node sends a message to the base stations via the LoRa radio protocol. One or more gateways receive the messages from the node and transmit them to the application through a network server. The gateways are also connected to a router responsible for setting their status and scheduling transmissions. Thus, each router is connected to one or more Brokers, which identify the sender node and send the message to the correct application, and vice versa, i.e., they choose the gateway to which they should send the application download messages. On the other hand, a Handler manages the application data, which is encrypted and decrypted in the Handler. On the other hand, the network server takes care of the LoRaWAN-specific functions [42]. In other words, the node joins the network through Over-The-Air Activation (OTAA) [54] and sends data to the created application. The gateway receives LoRa messages, converts them into information packets, and sends them to the router. The router receives the messages from the gateways and looks for a Broker to send them to. The Broker identifies the node, de-duplicates the traffic, and sends the packet to the Handler, who will register the application. The Handler encrypts and decrypts the payloads and publishes messages for the applications to the Brokers. Finally, the application collects and interprets the data.

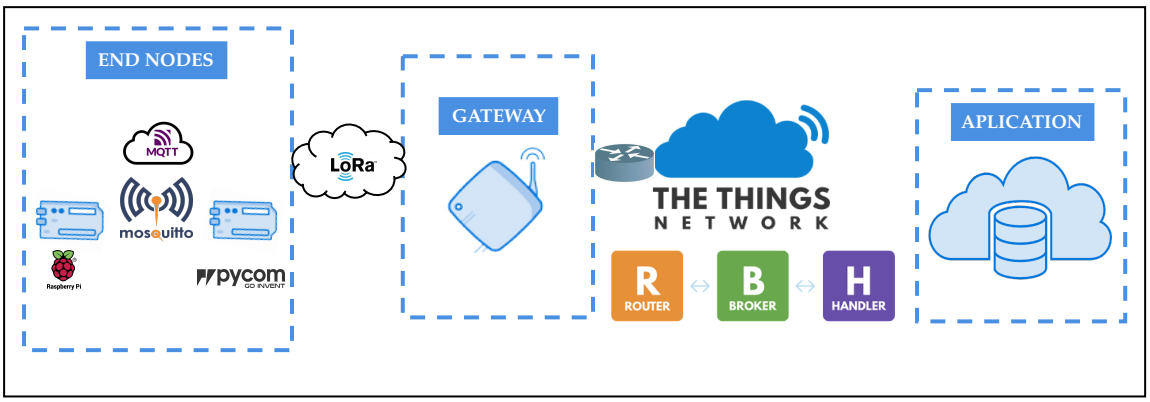


Figure 8. System communications architecture on the TTN platform.

2.1.4. LoRaWAN Link Processes

The working procedure of the system communications block can be seen in Figure 9. The code starts with the search and connects to a LoRaWAN network by the end node until it finds a LoRaWAN gateway that registers it. It then links Machine-to-Machine (M2M) communication [55,56] from the subscriber to the Broker over Wi-Fi. The Broker goes to a state of waiting for messages from the Publisher, then goes to image analysis and extraction of important information that represents the whole process to finally send this information to the Broker, which was waiting. This, in turn, transmits it through the LoRaWAN to a server in the cloud. So the user can see the data in the application.

The creation of the node, gateway, and application is described below.

The application is created in the TTN console to register the system node, for which it is registered with a user, and the application is selected. This application generates the Application Identifier (AppEUI), End-device Identifier (DevEUI), and Application Key (AppKey), which are added to the code inserted in Lopy. These keys are required to activate the node using OTAA [57], and the node attempts to join to send its data. Figure 10 shows the node activation scheme in OTAA mode. This OTAA mode is based on identifying the

end node and the target application by EUI (DevEUI and AppEUI) plus an encryption key used during negotiation (Appkey). In the join acceptance response, the server provides a device address (DevAddr), and both the end node and the server derive the network session key (NwkSKey) and the application session key (AppSKey) so that they never pass over the air.

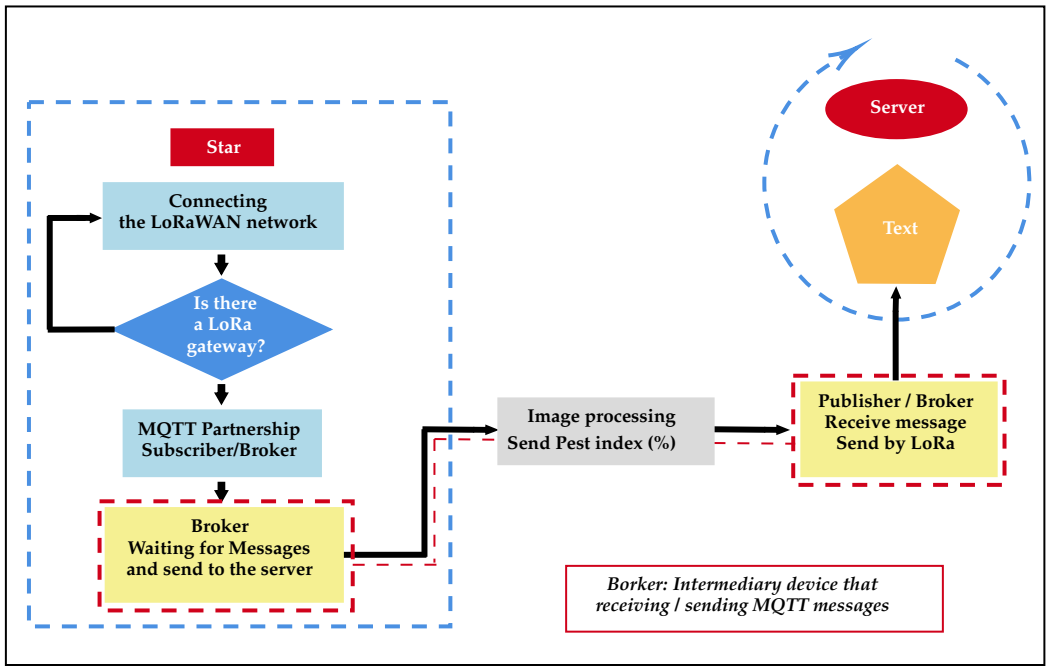


Figure 9. System transmission block flow diagram.

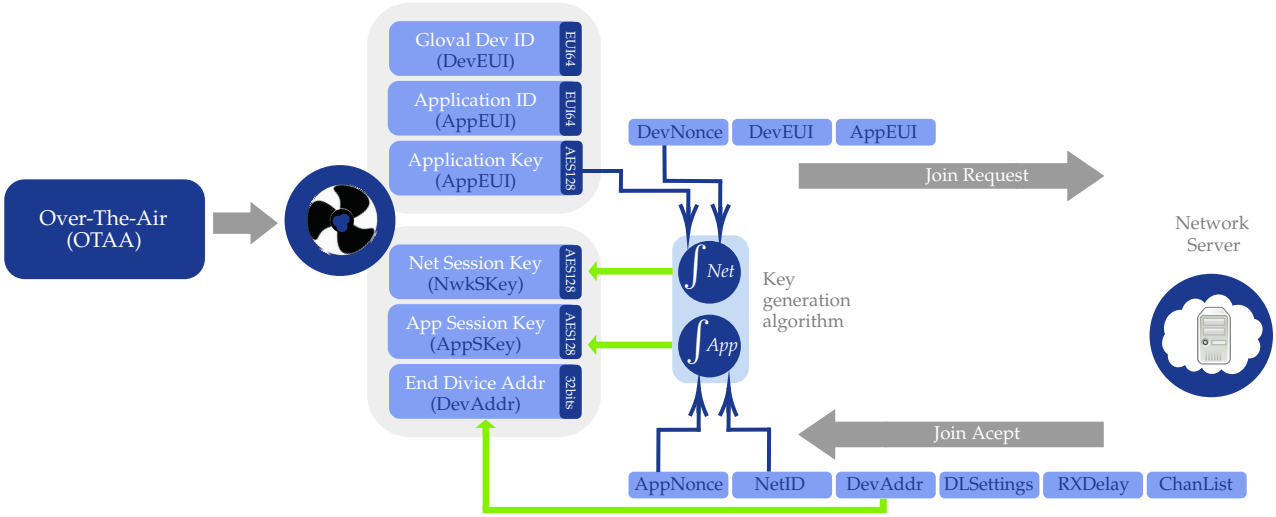


Figure 10. OTAA mode node activation schematic.

The Gateway device communicates the proposed LoRa-based system, and the application is created on the TTN platform.

The final node integration is a key part of the system's complexity and involves two key devices. The first is the Raspberry Pi, a versatile device that serves as an image analyzer and functions as an MQTT Broker/Publisher. The second device is the Pycom Lopy module, which acts as an MQTT client and is responsible for sending LoRaWAN messages via the LoRa radio module. This integration is a key part of the complexity of the system.

Working on the LoRa radio module of the microcontroller requires loading specific libraries such as MQTTclient, LoRa, and Pycom, and incorporating GitHub files [58]. These steps are key to configuring the LoRaWAN mode as a parameter for radio access and sending packets using LoRa modulation on the Europa 868 Mhz frequency. Also, the keys that the application generates are added to establish the OTAA connection, as shown in Figure 11.

```
# Initialize LoRa in LORAWAN mode.
lora = LoRa(mode=LoRa.LORAWAN)

# create an OTA authentication params
dev_eui = binascii.unhexlify('006A0653A9D4BC15'.replace(' ',''))
app_eui = binascii.unhexlify('70B3D57ED0010EFE'.replace(' ',''))
app_key = binascii.unhexlify('E9CE96E607D8DD003E491711D31D8D9C'.replace(' ',''))

# join a network using OTAA
lora.join(activation=LoRa.OTAA, auth=(dev_eui, app_eui, app_key), timeout=0)
```

Figure 11. Configuration parameters in LoRa radio module.

On the other hand, the MQTT Client parameters are added to the LoPy to subscribe to it, and these are the IP address of the MQTT Broker that, by default, the LoPy assigns the first IP to the Raspberry, 192.168.4.2. Then, an MQTT identifier and an MQTT Topic are created, so that the LoPy enters standby mode for MQTT messages.

Furthermore, the Paho-MMQT libraries, a significant tool, are used to configure the Raspberry Pi as a Publisher. After image processing, the messages can be published to the subscriber, which transmits them over the LoRa network. Figure 11 shows the use of functions and configuration parameters, highlighting the importance of these tools in our system. A key step in assigning the MQTT Broker function to the Raspberry Pi system is the installation and configuration of the Mosquitto software (version 1.4.10). This software is installed and configured to start automatically when the Raspberry Pi is powered on or rebooted, ensuring the system's smooth operation.

3. Results and Discussion

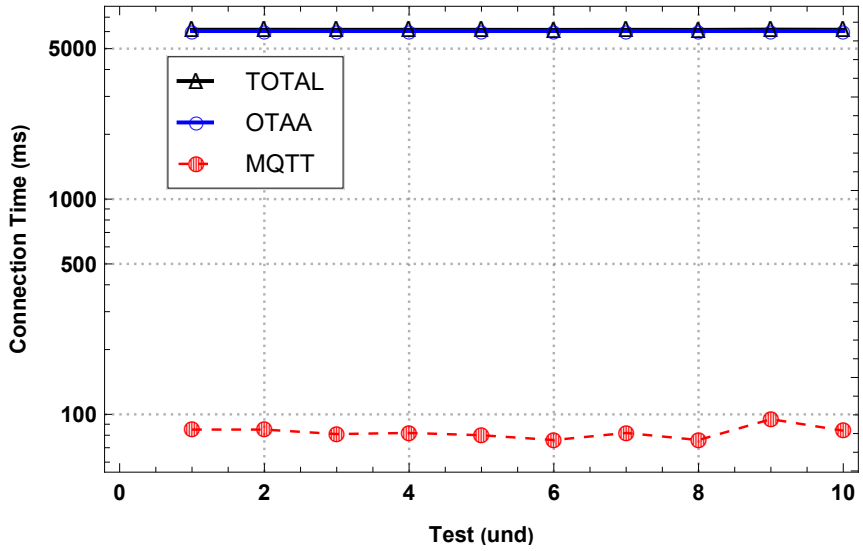
This section presents the main results obtained with the previous models. We present the results of the connection to the LoRaWAN network, the processing of the images containing the pest, and the data transmission times. On the other hand, for the image processing stage, 80 images had to be taken by the system, of which 40 images were chosen for the final analysis of the system due to the presence of several irregularities in the images that affected their processing. These irregularities were the poor capture of some images due to the presence of winds and instability of the prototype, capturing images that do not correspond to the area of interest, as shown in Figure 3a,b, which show a poorly captured image due to low illumination and a poorly captured image due to high illumination during capture. Numerous tests were carried out to obtain the results. However, ten tests (10 points on the x-axis) were taken to present the results.

Finally, the hardware, software, and tools used to design the proposed system model are as follows. We have used a Raspberry Pi 3 model B (Raspberry Pi Foundation, Cambridge, United Kingdom), which has a Broadcom BCM2387 chipset and an ARM Cortex-A53 processor with four cores as hardware. The RAM is 1 GB, and the graphics processor is VideoCore IV. A LoPy microcontroller (Pycom, Newcastle, Australia), whose size is 55 mm $\times$ 20 mm $\times$ 3.5 mm, also with a dual-core Espressif ESP32 chipset, 512 KB RAM, 4 MB external Flash memory, 802.1 b/g/n 16 mbps, Bluetooth Low Energy. In addition, regarding LoRa specifications, such as the Semtech LoRa SX1272 transceiver, power consumption with WiFi is 12 mA in active mode and 5 µA in standby mode, with LoRa being 15 mA in active mode and 1 µA in standby mode. A board for conditioning and operation. In addition, the pigtail, UFL connector, and SMA connector are used to connect to the LoRa antenna. A Logitech C920 Full HD webcam for capturing and analyz-

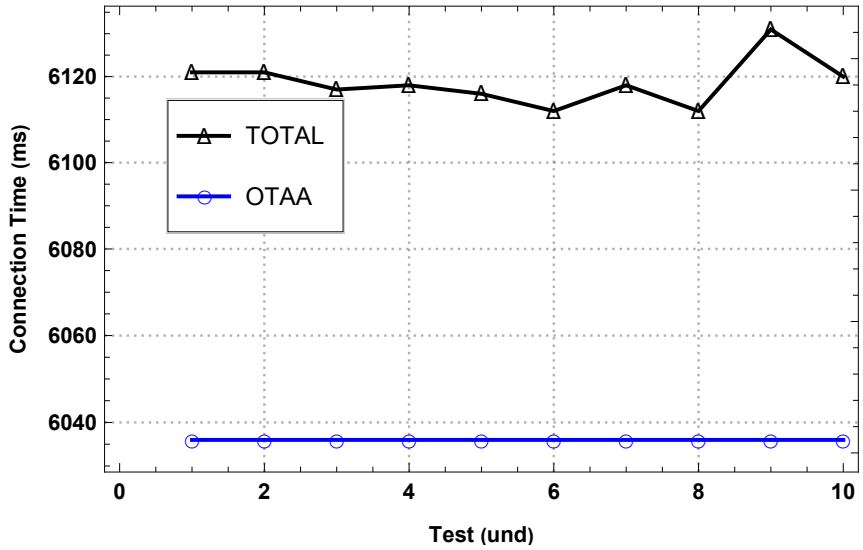
ing images. We have used MicroPython (version 1.9.6), a complete Python compiler that executes commands, as software to develop the system model; Open Source Computer Vision Library (OpenCV) for image processing; and Message Queuing Telemetry Transport (MQTT) libraries for compatibility with Pycom and Raspberry Pi for creating IoT-based projects. The final prototype is shown in Section 3.4.

3.1. Network Connection

In Figure 12, we can observe the behavior of the OTAA and MQTT times and the TOTAL time as a function of each test performed. Figure 12b shows that the OTAA time values remain at 6034 ms, indicating that the connection time to the LoRaWAN network takes 6034 s on average, which represents an advantage of the LoRaWAN protocol, while the MQTT time values remain at approximately 88 ms, indicating that the average connection time from the Lopy to the Raspberry Pi takes 88 ms. Finally, the TOTAL connection time to the node is the sum of the OTAA and MQTT times, so the TOTAL time is 6120 ms, as seen in Figure 12b.



(**a**)



(**b**)

Figure 12. Connection time in each test. (**a**) TOTAL, OTAA and MQTT times. (**b**) TOTAL and OTAA times.

3.2. Image Processing

3.2.1. Luminance and Cutting Time

The results of the image acquisition tests are divided into two parts. The first corresponds to high-resolution images (R = High), and the second to low-resolution images (R = Low), i.e., the first corresponds to 1920 × 1080 pixel images, and the second to 800 × 600 pixel images for each image accordingly.

Figure 13 presents a clear relationship between resolution and luminance, a measure of the brightness of an image. The average luminance of high-resolution images systematically exceeds that of low-resolution images, indicating that high-resolution images are brighter. This relationship underlines the impact of resolution on image quality. Furthermore, this luminance result is within the luminance range set by the system and maintains agreement for both types of images.

Figure 14 shows the capture time of the high and low-resolution images. We can observe that the capture time of a high-resolution image is longer than that of a low-resolution image. This result makes sense, as higher resolution equates to more data being captured and processed, which increases the capture time of this image compared to low-resolution images. However, it is important to note that the capture time for low-resolution images is not constant. This variability can lead to repeated requests for capturing images, which can be inefficient. Therefore, the most appropriate option is to capture higher-resolution images despite the longer capture time.

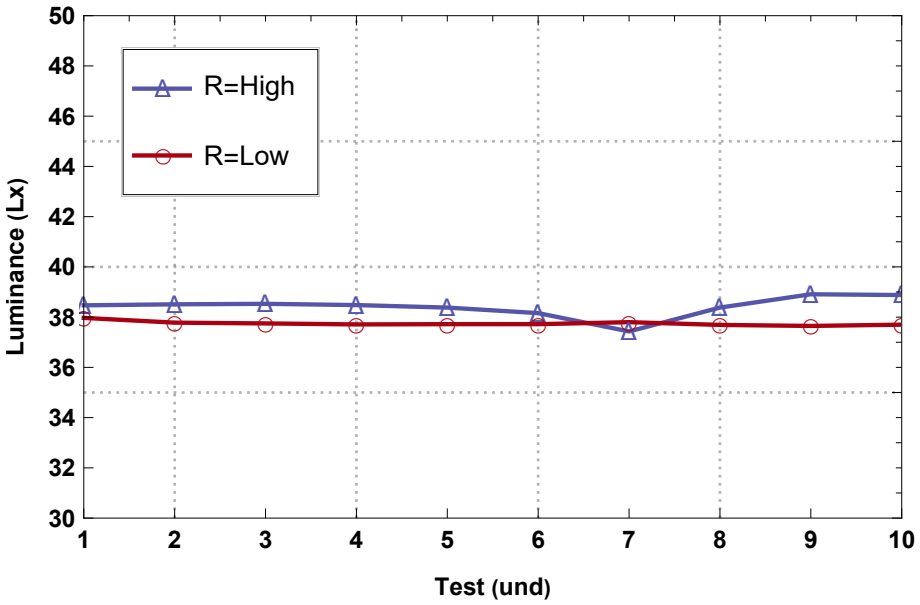


Figure 13. Luminance on each test.

3.2.2. Image Processing Steps

The captured images are subjected to RGB to HSV channel changes, dilation and smoothing, and morphological feature modifications such as segmentation and thresholding processes using the Raspberry Pi. The Raspberry Pi is a key component in our system, providing the necessary computational power for these image processing tasks. These processes isolate objects unrelated to the pest and the plant region of interest from the original image. The specific changes made by the Raspberry Pi are described below.

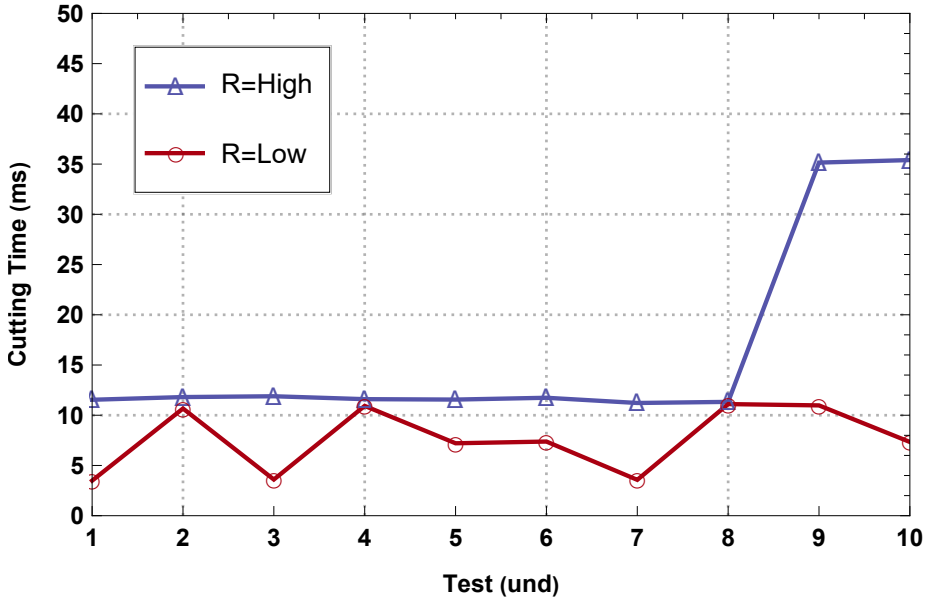


Figure 14. Cutting time depending on each test.

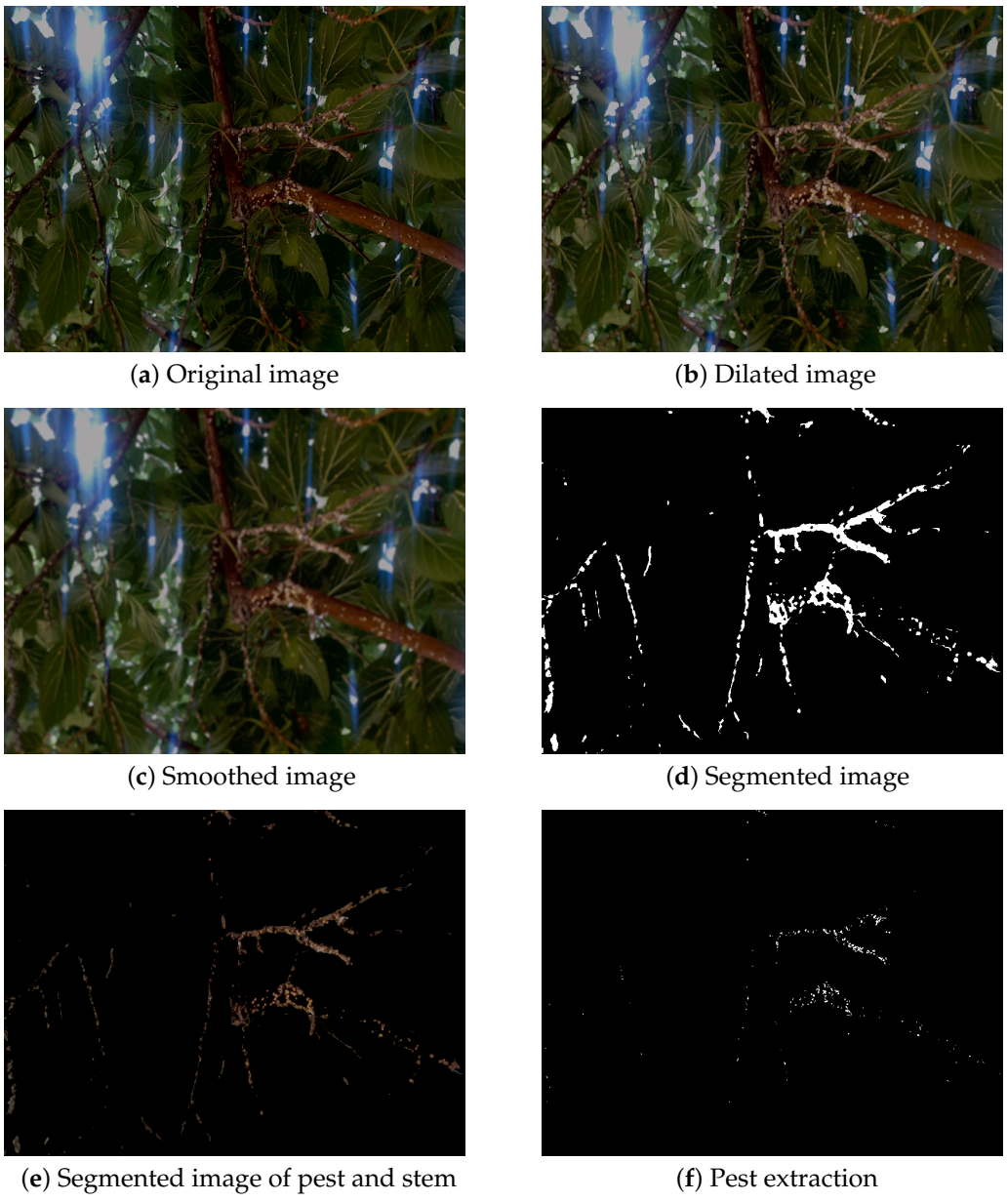
Figure 15a shows the original captured image of the plant and the pest. Figure 15b shows the transformation of the overall image after applying the dilation filter and highlighting the pest. Figure 15c shows the application of the Gaussian blur filter to the resulting image after applying the dilation. Figure 15d shows the application of the HSV filter to the resulting image after applying the Gaussian blur filter, resulting in the segmentation of the image using the HSV color space. The use of the HSV color space is particularly effective in this context as it separates the color information from the intensity information, making it easier to identify the pest. Figure 15e shows the application of the AND Filter Operator to the resulting image after applying the HSV filter. The result is an image highlighting the segmentation of the stem and the embedded pest. Figure 15f shows the separation of the pest, the removal of the brown range, and the binarization of the resulting image. Therefore, for the segmentation of the image, all the pixels that make up the objects belonging to the stem and the pest are obtained using the HSV color filter and thresholding. This method is the most suitable for the treatment of plants, as it allows the elimination of daylight, avoiding false positives. Since these false positives are interpreted as pests in the captured image.

On the other hand, the values of the RGB channels are adopted as selection criteria for both plants and pests. Although there are similarities in color between the stem and some leaves, the color of the pest can be identified and differentiated so that there are no restrictions on differences in size, volume, dominance, and shape, but only in phase, as this pest is initially white and changes to a yellow color as an adult. This color change represents greater reliability and preservation of the characteristics of the pest in our work, thus allowing the extraction of the referential percentage index of the pest found, which subsequently sends this information through the communications system.

3.2.3. Percentage Index of Pest Brightness and Image Processing Time

The percentage pest brightness index and image processing time results are obtained from the ten tests chosen from the total number of tests analyzed, varying the resolution of the acquired image. However, as the mean value of the resolution in the analyzed images was 5084 pixels in high quality, 5100 pixels were set as 100 percent. In addition, the time required in image processing has a higher capacity in recognizing the pest pixels. On the other hand, the percentage rates of pest brightness and their processing times for a variation of image resolution are as follows. L1 represents the percentage pest brightness index for an image with a resolution of 1920 $\times$ 1080 pixels per image, and T1 represents the time needed

to process this image. Based on the above, L2 and T2 represent the percentage index of pest brightness and the processing time for an image with a resolution of 1280 × 720 pixels, L3 and T3 represent the percentage index of pest brightness and the processing time for an image with a resolution of 1024 × 768, and L4 and T4 represent the percentage index of pest brightness and the processing time for an image with a resolution of 800 × 600 pixels.



(**a**) Original image (**b**) Dilated image

(**c**) Smoothed image (**d**) Segmented image

(**e**) Segmented image of pest and stem (**f**) Pest extraction

Figure 15. System image processing.

Figure 16 shows L1, L2, L3, and L4 as a function of each test. We can see that L1 > L2 > L3 > L4. This result shows that images captured at a higher resolution detect more pest pixels. Although the values of L4 vary throughout the tests, they show the highest percentage. In contrast, L2, L3, and L4 do not differ much in their values, but show percentages below 50% depending on the image resolution. We can also observe that L4 is below 50% in two tests. However, this is due to the winds that generate instability of the physical system, capturing an erroneous image position. In addition, it complies with the brightness conditions during the image capture process.

Figure 17 shows T1, T2, T3, and T4 as a function of each test. It can be seen that T1 is much higher than T2, T2 > T3 > T4. Furthermore, T1, T2, T3, and T4 maintain an

almost stable value for each test. This result shows that the processing time decreases when the image quality is reduced. However, this reduction in image resolution results in a loss of recognition of the pest pixels. Although T2, T3, and T4 are lower than T1, their corresponding L2, L3, and L4 do not exceed 50% and are considered unacceptable for transmitting information.

This result shows that images captured at a higher resolution detect more pest pixels. Although the values of L4 vary throughout the tests, they show the highest percentage, while L2, L3, and L4 do not vary in their values, showing lower percentages depending on the image resolution.

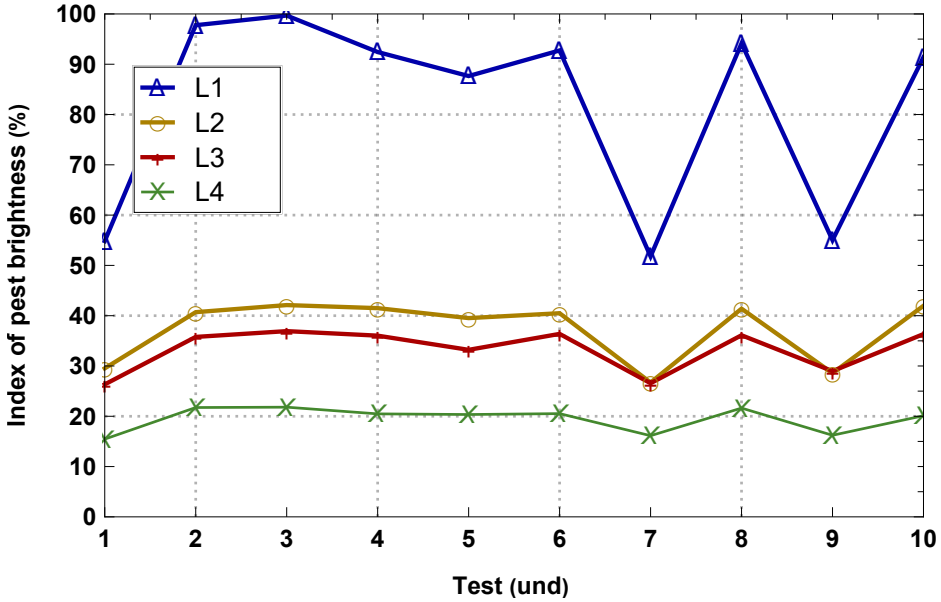


Figure 16. Percentage index of pest brightness in each test.

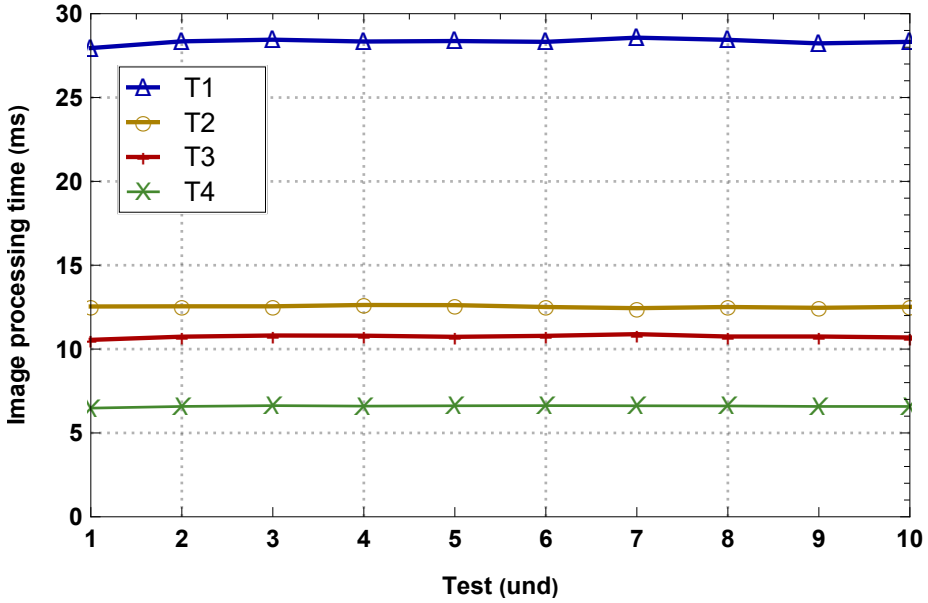


Figure 17. Image processing time in each test.

3.3. Message Transmission

Figure 18 shows the time taken to send messages to the LoRaWAN network in each test. It can be observed that the sending time presents a slight variation for each test performed, giving an average of 14.57 ms, which is within the acceptable range. This result

is due to the restrictions of using the LoRaWAN protocol infrastructure, such as the size of the packets to be sent not exceeding 255 bytes and the fact that messages cannot be sent at every instant. Therefore, the processing of the images is resolved at the node, and a representative value is extracted, such as the pest percentage index in the plant, which is approximately 17 bytes. On the other hand, in the transmission of messages by LoRaWAN, data are sent every 5 min for 2 h from the different images stored in the Raspberry Pi, and the message sending time is observed, which is 14.57 ms, as mentioned above.

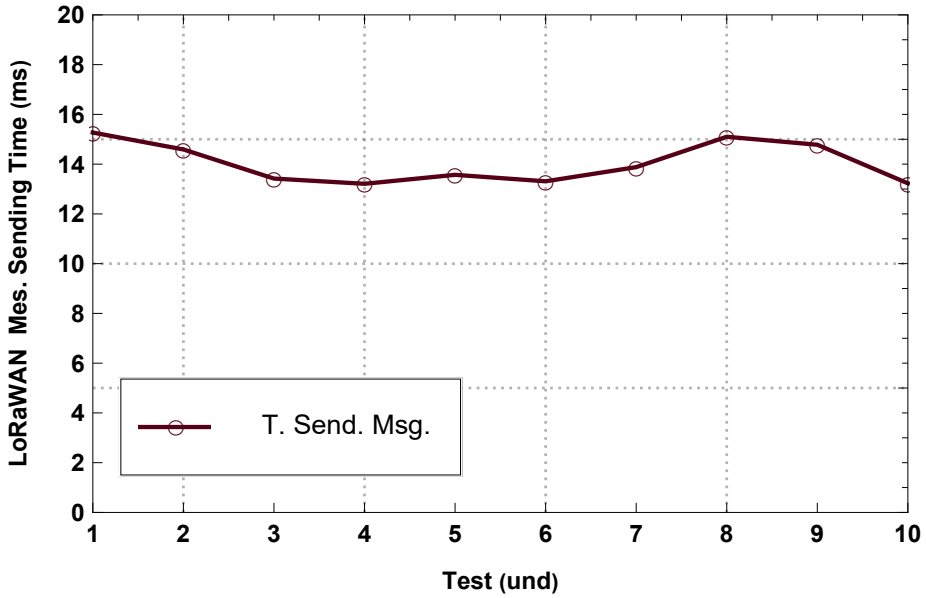


Figure 18. LoRaWAN message sending time.

3.4. Final Prototype

The final prototype developed and implemented for this study is shown in Figure 19.

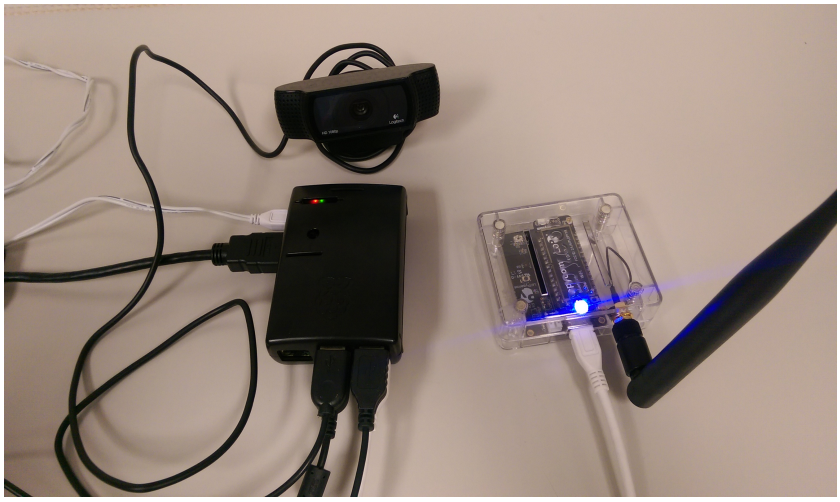


Figure 19. The final prototype integrates a Raspberry Pi small-board computer, LoPy microcontroller from Pycom and a Logitech C920 Full HD Webcam.

Based on all the results, we conclude that the average luminance condition for image acquisition is set to a value within the range of 36 to 42, with 38 being the most suitable, as the acquisition of images with high luminance produces false positives. Furthermore, the acquisition of images with low luminance allows computer vision techniques for image processing to be used appropriately. On the other hand, the results on image acquisition

time show that the better the quality of the captured image, the more pest pixels are detected. However, a lot of time is lost in image processing. Finally, for transporting messages based on the LoRaWAN protocol over the TTN platform, the transmission of messages prior to a connection is very short: 14 ms on average. However, the request and identification procedure increases the connection time.

4. Conclusions

This work has developed and evaluated the design of a communication prototype applied in precision agriculture. This design detects and monitors the Cotonet pest in crops by integrating advanced image processing technologies, wireless communication, and IoT platforms. This study was structured in four stages, each addressing a different aspect of the proposed system.

The first stage establishes the system model, combining tools such as Raspberry Pi and the Lopy microcontroller, as well as image processing and LoRa communication techniques. The choice of these tools and techniques makes it possible to address the specific challenges of precision agriculture, such as the need for real-time pest detection and monitoring in large agricultural fields. In the second stage, the lighting conditioning is optimized for image capture in varying outdoor conditions. The results show that lighting conditions, weather, and time of day influence the quality of the images captured. We identified clear days with sunlight and the period from 19:00 to 21:00 h of the day as the optimal conditions for image acquisition. These conditions improve image quality and processing accuracy. In the third phase, the communication infrastructure, namely the LoRaWAN and MQTT protocols, is analyzed in the context of the TTN network architecture. The implementation of these technologies facilitates efficient data transmission between the system nodes and the cloud application, despite the limitations of the LoRa protocol in terms of packet size and transmission frequency. Finally, the fourth step evaluates the performance of the detection system, including image registration, processing, and data communication. Analysis of the relationship between image resolution, brightness, and processing time shows that higher-resolution images (1920×1080 pixels per image) provide better detection of pest pixels (greater than 50% of the pest index) but require longer processing time (28.415 ms on average). This result emphasizes the need to balance image resolution with system operational requirements to optimize pest detection efficiency.

In conclusion, the developed system has proven effective pest detection and monitoring through a multidisciplinary approach that integrates image processing, wireless communication, and IoT technologies. However, the system faced several limitations, including image processing, where the tests were carried out under a particular scheme, i.e., this research was performed during the summer in a specific schedule where the behavior of luminosity was studied to achieve optimal results. Similarly, it includes the transmission of information since the LoRaWAN protocol has limitations regarding the size of the information frame sent over the channel, as our test bed was limited to sending packets smaller than 20 bytes, for which we did not analyze distances, bandwidth or energy consumption. Also, its current application is limited to the detection of a specific pest (Cotonet), and the extension of the system to other agricultural contexts requires further development. In summary, limitations include dependence on optimal lighting conditions, restrictions on communication protocols, measurement of power consumption, in-depth analysis of transmission parameters, and trade-offs between image resolution and processing time. Concretely, the results provide a solid basis for future research and improvements in the field of precision agriculture and in optimizing image capture technology and data transmission efficiency.

5. Future Work

This research opens new lines of study to implement more advanced and adaptive systems to overcome the current limitations in managing agricultural pest detection, i.e., to

improve the detection of pest pixels for higher-resolution images without requiring very long image processing times.

One of these new lines of study is detecting and monitoring pests using more advanced and adaptive technologies different from those used in this article. On the one hand, advanced techniques such as convolutional neural networks can be implemented to improve the accuracy of pest identification and image processing quality. On the other hand, adaptive systems can be developed that automatically adjust image capture according to variations in lighting and environmental conditions. In addition, future work could consider scalability and real-time application, such as developing adaptive algorithms that can adjust to dynamic changes in relation to lighting or incorporating multiple pest types into the detection model for wider agricultural environments. Finally, the system can be scaled for application in various crops and agricultural contexts, assessing its versatility and effectiveness. For example, the system can be adapted to identify aphids in vegetable crops or leaf spot diseases in cereals. For this, the system must be trained based on detecting specific images with these pests.

These lines of research have the potential to transform precision agriculture through the integration of technologies and multidisciplinary approaches. The continuation of this work will provide opportunities to address current constraints, such as limited accuracy in pest detection, contributing to more efficient and sustainable agriculture in the future.

Author Contributions: Conceptualization, methodology and software, J.L.Q.-V.; validation, J.L.Q.-V. and E.M.-C.; formal analysis, J.L.Q.-V.; investigation, J.L.Q.-V., E.M.-C. and E.J.S.-C.; data curation, J.L.Q.-V.; writing—original draft preparation, J.L.Q.-V., E.M.-C. and E.J.S.-C.; writing—review and editing, E.M.-C., E.J.S.-C. and Y.M.-C.; visualization, E.M.-C. and Y.M.-C.; supervision, E.J.S.-C.; project administration, J.L.Q.-V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Universitat Politècnica de València through the projects of the Networks and Computers Research Group (GRC) of the Department of System and Computer Informatics (DISCA) and partially by the Universidad Politécnica Salesiana under the Fog Computing Simulation project.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no potential conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PA	Precision agriculture
HSV	Hue Saturation Vue
IoT	Internet of Things
IP	Internet Protocol
LoRa	Long Range
LoRaWAN	Long-Range Wide-Area Network
M2M	Machine to Machine
MQTT	Message Queuing Telemetry Transport
OpenCV	Open Source Computer Vision Library
OTAA	Over-The-Air Activation
RGB	Red Green Blue
TTN	The Things Network

References

1. Saiz-Rubio, V.; Rovira-Más, F. From smart farming towards agriculture 5.0: A review on crop data management. *Agronomy* **2020**, *10*, 207. [CrossRef]
2. Soussi, A.; Zero, E.; Sacile, R.; Trinchero, D.; Fossa, M. Smart Sensors and Smart Data for Precision Agriculture: A Review. *Sensors* **2024**, *24*, 2647. [CrossRef] [PubMed]

3. Sharma, V.; Tripathi, A.K.; Mittal, H. Technological revolutions in smart farming: Current trends, challenges & future directions. *Comput. Electron. Agric.* **2022**, *201*, 107217.
4. Rahman, A.; Khalid, S.; Qadeer, S.; Naqqash, R.M.; Khalid, N. Cost-Benefit Analysis of Smart Pest Control Technologies. In *Revolutionizing Pest Management for Sustainable Agriculture*; IGI Global: Hershey, PA, USA, 2024; pp. 133–160.
5. Tudi, M.; Daniel Ruan, H.; Wang, L.; Lyu, J.; Sadler, R.; Connell, D.; Chu, C.; Phung, D.T. Agriculture development, pesticide application and its impact on the environment. *Int. J. Environ. Res. Public Health* **2021**, *18*, 1112. [CrossRef]
6. Sial, M.U.; Ahmed, R.; Khan, R.R.; Ahmad, J.N.; Riaz, A. Remote Sensing Technologies in Pest Detection. In *Revolutionizing Pest Management for Sustainable Agriculture*; IGI Global: Hershey, PA, USA, 2024; pp. 411–442.
7. Malik, J.A.; Saleem, M.; Almutairi, S.; Qadri, S.; Raza, M.A.; Alsanoosy, T.; Chaudhary, U.M.D. Implementing Fog Computing in Precision Agriculture for Real-Time Soil Health Monitoring and Data Management. In *Computational Intelligence in Internet of Agricultural Things*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 371–400.
8. Li, L.; Lin, J. Digital transformation for the sustainable development of firms: The role of green capability and green culture. *Sustain. Dev.* **2024**, *32*, 1861–1875. [CrossRef]
9. Sacoto-Cabrera, E.J.; Castillo, I.; Pauta, W.; Trelles, P.; Tamaríz, P.; Guambaña, L. Smart-water: Digital transformation of urban water measurement. In Proceedings of the 2022 IEEE ANDESCON, Barranquilla, Colombia, 16–19 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
10. Salam, A. *Internet of Things for Sustainable Community Development: Wireless Communications, Sensing, and Systems*; Springer Nature: Berlin/Heidelberg, Germany, 2024.
11. Bell, M.M. *Farming for Us All: Practical Agriculture and the Cultivation of Sustainability*; Penn State Press: University Park, PA, USA, 2024.
12. Chaudhari, B.S.; Zennaro, M.; Borkar, S. LPWAN technologies: Emerging application characteristics, requirements, and design considerations. *Future Internet* **2020**, *12*, 46. [CrossRef]
13. Růžička, J.; Sliacky, M.; Purkrábková, Z.; Hajčiarová, E. Opportunities of LoRaWAN Technology for Smart Cities-A Review. In Proceedings of the 2023 Smart City Symposium Prague (SCSP), Prague, Czech Republic, 25–26 May 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
14. Ramesh, R.; Arunachalam, M.; Atluri, H.K.; Kumar, C.; Anand, S.; Arumugam, P.; Amrutur, B. LoRaWAN for smart cities: Experimental study in a campus deployment. In *LPWAN Technologies for IoT and M2M Applications*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 327–345.
15. Correia, F.P.; Silva, S.R.d.; Carvalho, F.B.S.d.; Alencar, M.S.d.; Assis, K.D.R.; Bacurau, R.M. Lorawan gateway placement in smart agriculture: An analysis of clustering algorithms and performance metrics. *Energies* **2023**, *16*, 2356. [CrossRef]
16. Saban, M.; Bekkour, M.; Amdaouch, I.; El Gueri, J.; Ait Ahmed, B.; Chaari, M.Z.; Ruiz-Alzola, J.; Rosado-Muñoz, A.; Aghzout, O. A smart agricultural system based on PLC and a cloud computing web application using LoRa and LoRaWan. *Sensors* **2023**, *23*, 2725. [CrossRef] [PubMed]
17. Bonilla, V.; Campoverde, B.; Yoo, S.G. A Systematic Literature Review of LoRaWAN: Sensors and Applications. *Sensors* **2023**, *23*, 8440. [CrossRef]
18. Kornaros, G.; Bakoyiannis, D.; Tomoutzoglou, O. Smart Manufacturing Maintenance through LoRaWAN-based Ecosystem. In Proceedings of the 2023 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Dubrovnik, Croatia, 4–7 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 193–198.
19. Khan, A.A.; Laghari, A.A.; Awan, S.A. Machine learning in computer vision: A review. *EAI Endorsed Trans. Scalable Inf. Syst.* **2021**, *8*, e4. [CrossRef]
20. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine learning in agriculture: A review. *Sensors* **2018**, *18*, 2674. [CrossRef] [PubMed]
21. Chlingaryan, A.; Sukkarieh, S.; Whelan, B. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Comput. Electron. Agric.* **2018**, *151*, 61–69. [CrossRef]
22. Karunathilake, E.; Le, A.T.; Heo, S.; Chung, Y.S.; Mansoor, S. The path to smart farming: Innovations and opportunities in precision agriculture. *Agriculture* **2023**, *13*, 1593. [CrossRef]
23. Sishodia, R.P.; Ray, R.L.; Singh, S.K. Applications of remote sensing in precision agriculture: A review. *Remote Sens.* **2020**, *12*, 3136. [CrossRef]
24. Raj, E.F.I.; Appadurai, M.; Athiappan, K. Precision farming in modern agriculture. In *Smart Agriculture Automation Using Advanced Technologies: Data Analytics and Machine Learning, Cloud Architecture, Automation and IoT*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 61–87.
25. Huber, R.; Bartkowski, B.; Brown, C.; El Benni, N.; Feil, J.H.; Grohmann, P.; Joormann, I.; Leonhardt, H.; Mitter, H.; Müller, B. Farm typologies for understanding farm systems and improving agricultural policy. *Agric. Syst.* **2024**, *213*, 103800. [CrossRef]
26. Priya, N.M.; Amudha, G.; Dhurgadevi, M.; Malathi, N.; Balakrishnan, K.; Preetha, M. IoT and Machine Learning based Precision Agriculture through the Integration of Wireless Sensor Networks. *J. Electr. Syst.* **2024**, *20*, 2292–2299. [CrossRef]
27. Bongiovanni, R.; Lowenberg-DeBoer, J. Precision agriculture and sustainability. *Precis. Agric.* **2004**, *5*, 359–387. [CrossRef]
28. SS, V.C.; Hareendran, A.; Albaaji, G.F. Precision farming for sustainability: An agricultural intelligence model. *Comput. Electron. Agric.* **2024**, *226*, 109386. [CrossRef]
29. Alahmad, T.; Neményi, M.; Nyéki, A. Applying IoT sensors and big data to improve precision crop production: A review. *Agronomy* **2023**, *13*, 2603. [CrossRef]

30. Gonzalez, C.; Gibeaux, S.; Ponte, D.; Espinosa, A.; Pitti, J.; Nolot, F. An Exploration of LoRa Network in Tropical Farming Environment. In Proceedings of the 2022 IEEE 2nd International Conference on Computer Communication and Artificial Intelligence (CCAI), Beijing, China, 6–8 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 182–186.
31. Singh, R.K.; Aernouts, M.; De Meyer, M.; Weyn, M.; Berkvens, R. Leveraging LoRaWAN technology for precision agriculture in greenhouses. *Sensors* **2020**, *20*, 1827. [CrossRef]
32. Abu, N.; Bukhari, W.; Ong, C.; Kassim, A.; Izzuddin, T.; Sukhaimie, M.; Norasikin, M.; Rasid, A. Internet of things applications in precision agriculture: A review. *J. Robot. Control. (JRC)* **2022**, *3*, 338–347. [CrossRef]
33. Sacoto-Cabrera, E.J.; León-Paredes, G.; Verdugo-Romero, W. Lorawan: Application of nonlinear optimization to base stations location. In *Communication, Smart Technologies and Innovation for Society: Proceedings of CITIS 2021*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 515–524.
34. Swain, M.; Hashmi, M.F.; Singh, R.; Hashmi, A.W. A cost-effective LoRa-based customized device for agriculture field monitoring and precision farming on IoT platform. *Int. J. Commun. Syst.* **2021**, *34*, e4632. [CrossRef]
35. Arshad, J.; Aziz, M.; Al-Huqail, A.A.; Zaman, M.H.u.; Husnain, M.; Rehman, A.U.; Shafiq, M. Implementation of a LoRaWAN based smart agriculture decision support system for optimum crop yield. *Sustainability* **2022**, *14*, 827. [CrossRef]
36. Taşkin, D.; Yazar, S. A Long-range context-aware platform design for rural monitoring with IoT In precision agriculture. *Int. J. Comput. Commun. Control.* **2020**, *15*.. [CrossRef]
37. Tsouros, D.C.; Bibi, S.; Sarigiannidis, P.G. A review on UAV-based applications for precision agriculture. *Information* **2019**, *10*, 349. [CrossRef]
38. Cariou, C.; Moiroux-Arvis, L.; Pinet, F.; Chanet, J.P. Data collection from buried sensor nodes by means of an unmanned aerial vehicle. *Sensors* **2022**, *22*, 5926. [CrossRef]
39. Chamara, N.; Islam, M.D.; Bai, G.F.; Shi, Y.; Ge, Y. Ag-IoT for crop and environment monitoring: Past, present, and future. *Agric. Syst.* **2022**, *203*, 103497. [CrossRef]
40. Rasti, S.; Bleakley, C.J.; Holden, N.; Whetton, R.; Langton, D.; O'Hare, G. A survey of high resolution image processing techniques for cereal crop growth monitoring. *Inf. Process. Agric.* **2022**, *9*, 300–315. [CrossRef]
41. Ramesh, N.V.K. A Survey Of Agriculture Crop Monitoring Using IOT Based Image Processing And Machine Learning Techniques. *Int. J. Pharm. Res. (09752366)* **2020**, 435. [CrossRef]
42. Osorio, A.; Calle, M.; Soto, J.D.; Candelo-Becerra, J.E. Routing in LoRaWAN: Overview and challenges. *IEEE Commun. Mag.* **2020**, *58*, 72–76. [CrossRef]
43. Zhang, Y.; Guo, X.; Ma, J.; Liu, W.; Zhang, J. Beyond brightening low-light images. *Int. J. Comput. Vis.* **2021**, *129*, 1013–1037. [CrossRef]
44. Alzahrani, A.; Aldhyani, T.H. Artificial intelligence algorithms for detecting and classifying MQTT protocol Internet of Things attacks. *Electronics* **2022**, *11*, 3837. [CrossRef]
45. Mazon-Olivo, B.; Hernández-Rojas, D.; Maza-Salinas, J.; Pan, A. Rules engine and complex event processor in the context of internet of things for precision agriculture. *Comput. Electron. Agric.* **2018**, *154*, 347–360. [CrossRef]
46. UBUNTU. UBUNTU. 2023. Available online: http://manpages.ubuntu.com/manpages/cosmic/man1/v4l2-ctl.1.html (accessed on 1 June 2023).
47. Ikeuchi, K. *Computer Vision: A Reference Guide*; Springer: Berlin/Heidelberg, Germany, 2021.
48. Alegre, E.; Pajares, G.; de la Escalera, A. *Conceptos y Métodos en Visión por Computador*; Grupo de Visión del Comité Español de Automática (CEA): Barcelona, Spain, 2016.
49. Guerrero Hernández, J.M. *Sistema de Visión Para Agricultura de Precisión: Identificación en Tiempo Real de Líneas de Cultivo y Malas Hierbas en Campos de Maíz*; Universidad Complutense de Madrid: Madrid, Spain, 2015.
50. Marcelino, R.; Casagrande, L.C.; Cunha, R.; Crotti, Y.; Gruber, V. Internet of Things applied to precision agriculture. In Proceedings of the Online Engineering & Internet of Things: Proceedings of the 14th International Conference on Remote Engineering and Virtual Instrumentation REV 2017, Columbia University, New York, NY, USA, 15–17 March 2017; Springer: Berlin/Heidelberg, Germany, 2018; pp. 499–509.
51. OpenCV. Image Filtering. 2023. Available online: https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=gaussianblur#gaussianblur (accessed on 1 June 2023).
52. Rojas Gómez, R.A. *Diseño de un Sistema de Registro de Imágenes Orientado a la Agricultura de Precisión;* Pontificia Universidad Católica del Perú: San Miguel, Perú, 2011.
53. Prado Liceaga, M.D. *Diseño y Despliegue de Arquitectura Para la Recogida y Presentación de Medidas de Sensores IoT*; Universidad del País Vasco: Bilbao, Spain, 2017.
54. Rocha, L.A.; Barreto, F.; Seman, L.O. The Internet of Things LoRaWAN Technologies in Academia: A Case Study. *The Internet of Things in the Industrial Sector: Security and Device Connectivity, Smart Environments, and Industry 4.0*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 193–219.
55. Amodu, O.A.; Othman, M. Machine-to-machine communication: An overview of opportunities. *Comput. Netw.* **2018**, *145*, 255–276. [CrossRef]
56. Weyrich, M.; Schmidt, J.P.; Ebert, C. Machine-to-machine communication. *IEEE Softw.* **2014**, *31*, 19–23. [CrossRef]

57. Borsos, D. Development and Optimization of Exit Procedures for LoRaWanNodes. In Proceedings of the 2024 IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY), Pula, Croatia, 19–21 September 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 000523–000527.
58. Pycom. Pycom. 2023. Available online: https://forum.pycom.io/topic/429/otaa-connection-to-ttn (accessed on 1 June 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.