2024 33rd International Conference on Computer Communications and Networks (ICCCN) | 979-8-3503-8461-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCCN61486.2024.10637527

iCrop: Enabling High-Precision Crop Disease Detection via LoRa Technology

Xu Tao* Jackson Butcher* Simone Silvestri* Flavio Esposito‡
*Department of Computer Science, University of Kentucky, USA
‡Department of Computer Science, Saint Louis University, USA
Invited Paper

***Abstract*—Crop disease recognition is a fundamental keystone in enabling disease control, limiting disease spread, and mitigating farmers' losses. Recently, advanced image processing techniques for crop disease detection, based on deep learning, have gained significant popularity. However, the practical deployment of these models in real farms remains challenging. This is mostly due to the lack of Internet connectivity which prevents the transmission of the acquired images to sufficiently powerful edge/cloud servers to execute such complex models. LoRa has emerged as a promising network solution for rural areas, thanks to its extensive communication range and cost-efficient deployment. However, the low data rate of this technology prevents its effective application for the transmission of large images for crop disease detection. In this paper, we propose a LoRa-based framework called *iCrop*. iCrop enables high disease classification accuracy while exploiting the cost-effectiveness of LoRa transmission technologies. Specifically, iCrop is based on a *LoRa Node*, which captures crop leaf images and pre-processes them through image segmentation. The node selects and transmits the most informative segments over LoRa to the *LoRa Edge Server*. The server, in turn, runs the disease classification using a Convolutional Nerual Network (CNN) deep learning model empowered with majority voting among segments. To prevent data losses, typical of LoRa transmission, we develop a reliable transmission protocol on top of LoRa, which takes care of retransmissions and efficient communication. Extensive experiments on a real LoRa testbed show the advantages over two comparison approaches with respect to several performance metrics.**

***Index Terms*—LoRa, Crop Disease Monitoring, Smart Farming, Precision Agriculture.**

I. Introduction

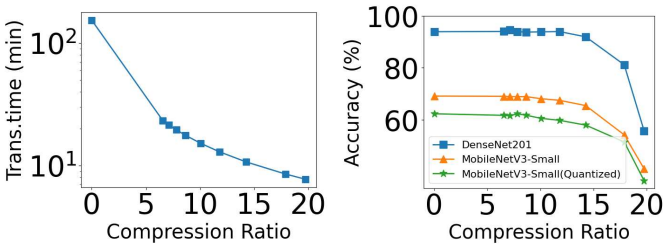
The persistent threat of crop diseases poses a significant challenge to crop productivity, quality, and farmers' economic stability [1] [2]. *Early and accurate* detection of diseases is pivotal to effectively controlling their spread and mitigating losses [3]–[5]. With the rapid developments in remote sensing [6], digital imaging [7], and deep learning [8], advanced image processing techniques for crop disease detection have gained significant popularity. These techniques replace traditional time-consuming visual inspections and symptom-based identification approaches [9]–[11].

Numerous research efforts have concentrated on enhancing model performance to attain superior disease classification accuracy. Notably, convolutional neural networks (CNN) have demonstrated exceptional capabilities in image processing and classification [11]–[13]. Despite these advancements, the practical deployment of these models for fully automating disease detection in real farms remains a challenge. Several solutions have proposed training Deep Neural Network (DNN) models with extensive datasets of manually captured crop leaf images in controlled environments [11]–[13]. However, utilizing such methods in practice necessitates on-site presence of farmers for image acquisition, resulting in *labor-intensive*, *costly*, and *time-consuming* monitoring.

To mitigate such challenges, alternative approaches have been proposed, leveraging ground robots [14] or unmanned aerial vehicles (UAVs) [15], [16] to automatically acquire crop field images. Despite the potential advantages, these approaches are limited by the *connectivity bottleneck* arising when transmitting large images to a remote (edge) server for analysis and classification. This bottleneck is due to the challenge of providing connectivity in rural areas and the large scale of such rural fields [17]. These make short-range connectivity solutions, such as WiFi and Bluetooth, inadequate [18]. Often, 4G/5G services are also not a viable option for lack of coverage [19]. This is because, despite the significant government efforts to improve rural connectivity, such as in the USA [20], there is still a lack of pervasive deployment in remote areas [18]. Furthermore, these connectivity solutions often hide unbearable expenses to be extensively adopted by farmers. To address these connectivity issues, recent efforts have focused on *on-device AI*, where machine learning models are executed on end-devices to reduce the amount of transmitted data. As an example, approaches include model compression and quantization techniques, to facilitate the deployment of deep neural networks on low-powered end-devices [21], as well as TinyML solutions [22]. Despite advancements in this direction, on-device AI still has notable challenges, including memory restrictions for image data processing and low accuracy with hard samples [23]. Consequently, as of today, data processing using complex deep learning models at more powerful servers remains necessary.

Long Range (LoRa) [24], has emerged as a promising network solution for remote areas communications. Its exceptional abilities to provide extensive communication range and cost-efficient deployment, make it the most promising solution to tackle network coverage challenges in large-scale rural farms. This, consequently, facilitates the development and

Jackson Butcher is an undergraduate student in the Computer Science Department at the University of Kentucky, USA.



(a) (b)

Fig. 1. Transmission time vs compression ratio (a) and Crop disease classification accuracy vs compression ratio with three different models (b). Reducing the image size can decrease the transition time over LoRa but results in a significant drop in crop disease classification accuracy even with a complex CNN model.

deployment of precision agriculture solutions [25]. However, existing works have primarily utilized LoRa for transmitting small sensor scalar data such as temperature, humidity, carbon dioxide levels, electrical conductivity, and illuminance [26]–[28]. This is primarily because LoRa allows a longer communication range at the cost of a considerably narrow bandwidth. As an example, even when utilizing the high data rate setting with a spreading factor 7 and a bandwidth of 250KHz, the resulting data rate remains only at approximately 5.7Kbit/s. Consequently, precision agriculture applications that require transmitting images, such as crop disease recognition and insect detection, are still not benefiting from LoRa communications and on-device AI solutions [29].

Motivating Example. To show an example of LoRa's limitations in transmitting images and enabling effective precision technology applications, in Figure 1 we report some preliminary experiments to motivate our work. Here, we transmitted over LoRa an RGB image of a soybean leaf with a size of 5MB from the Auburn Soybean Disease Image Dataset (ASDIAD) [30]. We also adopt DenseNet201-based Convolutional Neural Network (CNN) classifier proposed in [31]. Additionally, we implement two TinyML models as on-device AI solutions. Specifically, we adopt the MobileNetV3-Small, proposed in [32], and a quantized version of MobileNetV3-Small, further reducing its size.

In order to reduce the amount of transmitted and processed data, we show the transmission time for a single image (Figure 1 (a)) and the overall classification accuracy (Figure 1 (b)), versus different Joint Photographic Experts Group (JPEG) image compression factors. A compression factor of zero corresponds to the original image. As the figure shows, it takes more than 100 minutes to transmit over Lora a single RGB image. Higher level of compression allows to significantly reduce the image size, and thus transmission time and the computational burden for the end-device, but also incurs the risk of a significant drop in classification accuracy. The best tradeoff for DenseNet201-based, in this example, is a compression factor of 13, which is able to provide an accuracy above 90%. However, this still requires more than 13 minutes to transmit a single image, clearly not scaling well with the hundreds of images required to monitor an agricultural field. The TinyML models results in a notable decrease in accuracy compared to DenseNet201 for the soybean disease dataset, as depicted in Figure 1 (b). This decline can be attributed to the inherent challenge faced by TinyML in handling complex samples, exacerbated as image size reduces.

Contributions. In this work, we propose a LoRa-based autonomous framework, called *iCrop*. iCrop enables crop disease classification by exploiting the cost-effectiveness of LoRa transmission technologies while providing *high disease classification accuracy*. Figure 2 shows an overview of iCrop. The *LoRa Node* is an end computing device to be deployed in the field, such as Raspberry Pi or Microcontroller Unit (MCU) equipped with LoRa capability, which captures and transmits crop leaf images. To minimize the data transmission time, we propose an efficient image pre-processing technique suitable for such end devices. Specifically, we resort on *image segmentation* and we use an efficient approach to identify informative segments within the original image. Only a few informative segments are selected, compressed, and then transmitted over LoRa. The *LoRa Edge Server*, equipped with sufficient computing resources, performs the CNN deep learning model training and testing for crop disease recognition. Upon receiving image segments, the pre-trained deep learning model generates disease predictions for each segment. Subsequently, a majority voting mechanism consolidates these predictions, yielding the final disease diagnosis.

Data loss happens frequently over LoRa networks due to factors such as radio signal interference or obstacles. Since the selected segments contain the most informative disease features, any additional loss could decrease the accuracy. Therefore, we also propose a reliable communication protocol, running on top of LoRa, to ensure the receipt of all the information within the selected segments.

We develop a real LoRa testbed to validate the proposed approach. Our experiments demonstrate that our approach yields performance improvements of up to eightfold when compared against two alternative methods across various performance metrics.

In particular, the main contributions of this paper are:

- We present an end-to-end autonomous architecture for crop disease detection, leveraging LoRa-based image transmission and pairing it with the CNN-based deep learning techniques.
- We propose an optimized image pre-processing approach, designed for implementation on end devices and low-data rate communication technologies, to minimize data transmission time while achieving a high crop disease classification accuracy.
- We introduce a reliable transmission protocol on top of LoRa communication to handle the data loss caused by radio interference or obstacles.
- We implement a lab-scale testbed, and conduct extensive experiments to validate the performance of the proposed framework against several LoRa metrics, i.e., spreading factor, bandwidth, and distance.

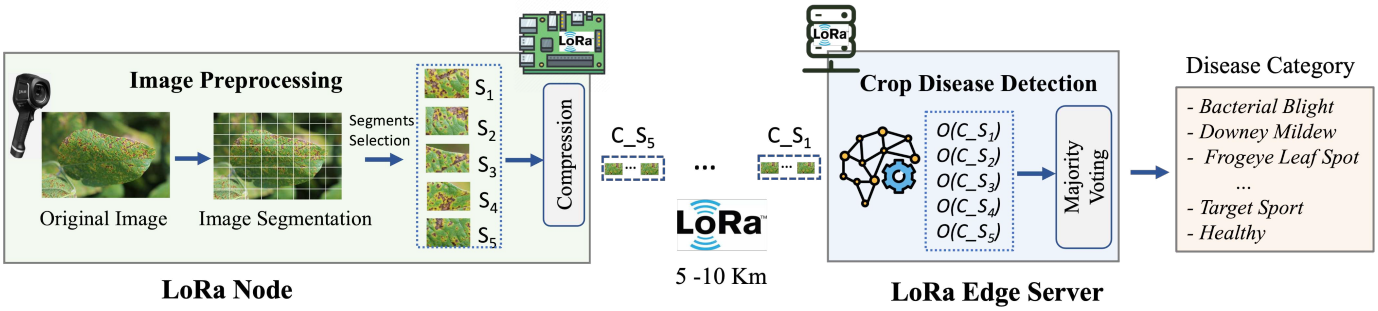


Fig. 2. The proposed end-to-end architecture for autonomous crop disease detection.

The rest of the paper is organized as follows. In Section II, we present the architecture of iCrop, including the proposed image preprocessing method, and the reliable communication protocol. Section III describes the implementation of the lab scale testbed and presents the performance results of the evaluation conducted with the testbed. SectionIV provides a literature review and identifies the research gaps of the existing works. Finally, conclusions are drawn in Section V.

II. The Proposed Architecture: iCrop

In this section, we present iCrop, our end-to-end architecture for automated crop disease detection based on low-cost image transmission over LoRa. iCrop can significantly reduce the amount of data transmission while offering high accuracy for crop disease recognition.

As depicted in Figure 2, our system comprises two primary components: the *LoRa Node* and the *LoRa Edge Server*. The LoRa Node, serving as an end-device deployed in the crop field, is powered by battery[1]. It facilitates LoRa transmission and offers limited computing capabilities. This node can be constructed using a Raspberry Pi or MCU, integrated with a LoRa transceiver module for LoRa communication and an RGB camera to capture images of crop leaves. The LoRa node could also be mounted on a robot scouting the field [33]. On the other hand, the LoRa Edge Server is realized with a LoRa transceiver to receive data transmitted by the LoRa Node. It is equipped with more capable computing resources, e.g., a laptop, capable of supporting the training and testing of deep learning-based image processing models for crop disease classification.

A. Image Preprocessing in LoRa Node

Crops affected by diseases often exhibit noticeable symptoms, such as discoloration, lesions, and spots. Image processing techniques generally identify crop diseases by extracting key features such as leaf color, texture, and shape patterns, from parts of the original image. This implies that transmitting every detail of the full image is unnecessary and results in excessively long transmission time, as illustrated in Figure 1. While compression can reduce the transmitted data, relying on sole compression results in images that still contain significant redundant information, such as the background, prolonging the transmission time over LoRa.

To enhance the efficiency of image transmission, we introduce an efficient image pre-processing method referred to as *Top K Segments Selection*, executed by the LoRa node. This method first divides the image into *segments* by splitting the image into an $n \times n$ grid. We denote each of the segments as S_i. Figure 3 shows an example of an 8×8 segmentation of a soybean leaf afflicted with bacterial blight disease.

Next, we want to identify the most informative K segments to transmit, where K is a predefined parameter. Intuitively, our method is based on the realization that segments with the most colors contain more information. As a result, for each segment S_i, we compute the set of colors $\mathcal{T}_i$. Specifically, consider a pixel $p_j = (r, g, b)$, where (r, g, b) are the red, green, and blue channel values for p_j. For each $p_j \in S_i$, we insert the tuple in $\mathcal{T}_i$, treating is as a mathematical set with no repetitions. As a result, we define the informativeness w_i of segment S_i as the cardinality of $\mathcal{T}_i$, i.e. $w_i = |\mathcal{T}_i|$. Figure 3 also shows a value of w_i for each segment in the image.

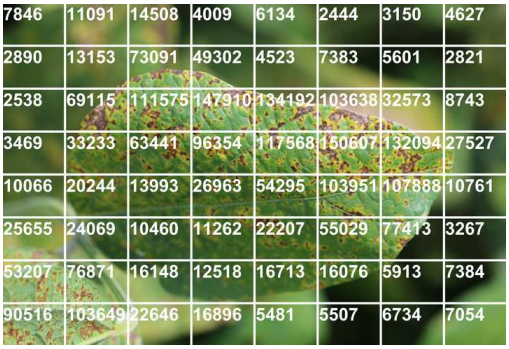


Fig. 3. A segmentation example of a soybean leaf image affected by bacterial blight disease and the assigned weight.

The next step of the segment selection picks the top K segments with the highest weight. Figure 4 shows the selected segments from Figure 3 with $K = 5$. Clearly, this simple method is able to select the most informative segments.

Once the most informative segments are identified, these are compressed using JPEG compression. Here we adopt

[1]Energy-harvesting-based solutions can also be potentially employed to power the LoRa Node.

a compression factor of 10. The compressed segments are subsequently transmitted via LoRa.

Fig. 4. Segments selected by Top $K = 5$ Segments Selection.

B. Reliable Transmission Protocol over LoRa

Upon the selection and compression of the most informative segments, the LoRa node initiates the transmission of these segments to the LoRa Edge Server. To enhance the transmission efficiency while adhering to bandwidth constraints, we employ a strategy of dividing each segment into smaller chunks, subsequently encapsulating them within data packets as shown in Figure 5. Each data packet has a length of 64 bytes. The first two bytes represent the packet ID, PID, while the remaining 62 bytes are the payload and contain the encoded pixels of the segments. The advantage of relying on packets lies in more efficient retransmission in case of packet loss, since these are of much smaller size than segments.

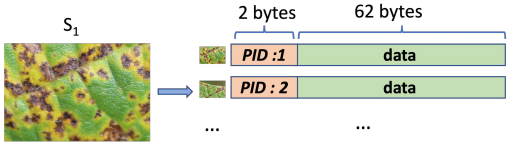


Fig. 5. An example of segment split into small packets of length 64 bytes.

LoRa operates within unlicensed frequency bands [25], making it vulnerable to interference and collisions, potentially resulting in data loss. This is exacerbated by the constraints of limited bandwidth and transmission power. Given that the selected segments encapsulate vital information, any omission could compromise the accuracy of predictions. To address this concern, we design a reliable protocol layered on top of LoRa communication. This protocol efficiently manages packet loss by implementing a mechanism for retransmitting missing packets, as depicted in in Figure 6.

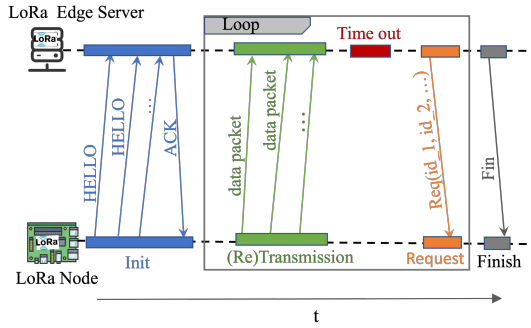


Fig. 6. The proposed transmission protocol on top of LoRa communication.

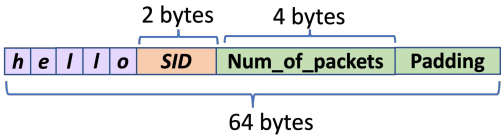


Fig. 7. The structure of $HELLO$ packet.

The transmission protocol works as follows. Initially, the LoRa Node sends *HELLO* packets, to alert the LoRa Edge Server of the intent of transmitting a segment. The structure of the *HELLO* packet is depicted in Figure 7. These packets also have a length of 64 bytes and include an "hello" label, followed with SID of two bytes representing the segment ID. The SID is followed by the $Num\_of\_packets$ of four bytes, indicating the number of packets within that segment, and finally a padding[2].

Upon receipt of a *HELLO* packet, the LoRa Edge Server responds with an *ACK* containing the SID. Subsequently, the LoRa Node starts sending the data packets of the segment SID to the LoRa Edge Server. The LoRa Edge Server receives and stores the packets. When no more packets are received for a specified time period, the LoRa Edge Server assumes that the LoRa nodes has completed the transmission and it checks if all packets have been received correctly. If some packets are lost or compromised, the LoRa Edge Server requests missing packets from the LoRa Node, specifying the packet PIDs. Upon receiving the request, the LoRa Node retransmits the missing packets. This procedure is repeated until all packets for that segment are received by the LoRa Edge Server. Finally, the LoRa Edge Server signals the succesful receiption of all packets for the current segment by transmitting a Fin. The LoRa Edge Server then reconstructs the segment by assembling all the received packets.

C. CNN-based Model for Crop Disease Classification

The CNN model for crop disease classification is built upon DenseNet201. The architecture is depicted in Figure 8. Specifically, we extend the model introduced in [31]. To this purpose, the extended model works with segments obtained through the image pre-processing method on the LoRa Node, treating each segment within the image as an individual input. For the CNN model, we employ a transfer learning approach. Specifically, DenseNet201 is initialized with the pre-trained ImageNet weights, and its layers are utilized for feature extraction. The last fully-connected layer is removed, and a global average pooling layer is added to reduce the number of parameters. Subsequently, a fully connected layer comprising 128 neurons with a ReLU activation function and a dropout rate of 0.2 is incorporated. This is followed by another fully connected layer with 64 neurons, a ReLU activation, and a dropout rate of 0.3. A softmax layer is then added to facilitate the eight-disease classification for each segment.

[2]Our packets are encapsulated into LoRa packets, which handle framing and transmission errors through checksum.

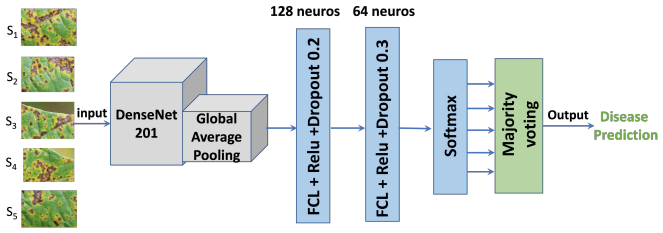


Fig. 8. Proposed CNN-based crop disease classification model.

To determine the final crop disease class of each crop image, a majority voting layer is integrated. The ultimate prediction is made by selecting the label with the highest occurrence of the K segments selected by the Lora Node. For instance, with $K = 5$, if the outputs are $\{$*Bacterial Blight, Bacterial Blight, Target Spot, Bacterial Blight, Target Spot*$\}$, the final output will be *Bacterial Blight*.

III. Evaluation

In this section, we first discuss the LoRa testbed developed to evaluate iCrop. Then, we compare our approach with two benchmarks to demonstrate its superior performance.

A. Testbed Implementation

We implement the LoRa Node using a Raspberry Pi 3 model B [3] [34], depicted in Figure 9. A touch screen is added for visualization purposes and enhancing user interface interactions. Furthermore, we equip the Raspberry Pi with an RGB camera, to capture images and with a Heltec WiFi LoRa 32 V2 transceiver [35], to provide the LoRa communication capabilities. LoRa transmits over license-free megahertz radio frequency bands 915MHz (North America). The communication between the LoRa transceiver and the Raspberry Pi is established through serial communication.

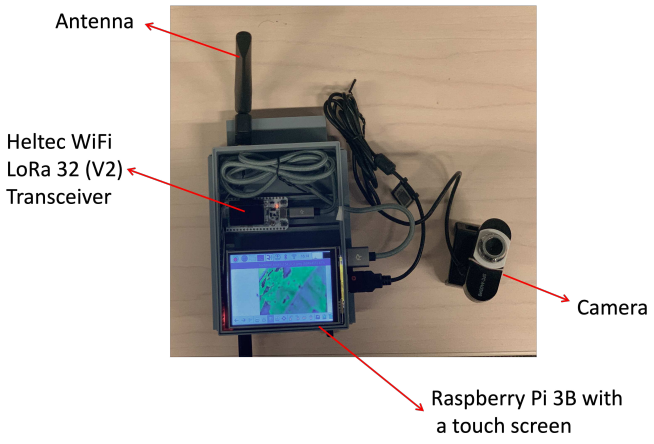


Fig. 9. The LoRa Node: a Raspberry Pi, with an RGB camera, touch screen, and a Heltec WiFi LoRa 32 V2 transceiver.

[3]The LoRa Node can also be implemented with a more energy efficient device such as MCU.

The implementation of the LoRa Edge Server is shown in Figure 10. We use a Republic of Gamers laptop, to support the CNN-based crop disease inference and mount the same LoRa transceiver as in the LoRa Node to receive data. The specs of the LoRa Edge server are summarized in Table I.

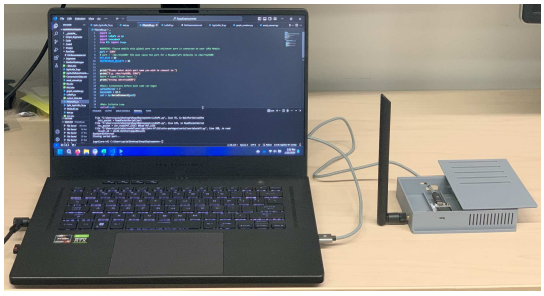


Fig. 10. LoRa Edge Server: a Republic of Gamers laptop and a Heltec WiFi LoRa 32 V2 transceiver.

TABLE I
The Configuration of the LoRa Edge Server

Model	ROG Zephyrus G15 GA503RS_GA503RS
Processor	AMD Ryzen 9 6900HS with Radeon Graphics, 3301 Mhz, 8 Core(s), 16 Logical Processor(s)
GPU	NVIDIA GeForce RTX 3080 Laptop GPU
RAM	16GB

B. Benchmarks

The first comparison corresponds to the transmission of full original image. The classification is carried out with the CNN approach proposed in [31]. We denote this approach as *Full*. This approach can provide the best accuracy in crop disease classification, at the expense of an extensive transmission time.

The second benchmark is based on the transmission of JPEG compressed images. In this case, the LoRa Node compresses the full image and transmits it to the LoRa Edge Server. Using our preliminary experiments in Figure 1, we opt for a compression ratio of 13, since it represents the best tradeoff between high accuracy and transmission time. Also in this case, the classification is carried out with the CNN approach proposed in [31], but the input images are the images compressed by the LoRa Node. We denote this approach as *JPEG13*.

Note that, for fairness of comparisons, we adopt our transmission protocol also for the benchmark comparisons. As a result, all approaches benefit from reliable communication and are able to transmit the images successfully, although with different transmission times. We discuss the dataset, training and testing of the classification models, in the next section.

C. Experiment Setup

1) Dataset: We utilize the publicly available Auburn Soybean Disease Image Dataset (ASDIAD) [30]. This dataset consists of $9,648$ images acquired during the 2020 and 2021 soybean seasons using a digital SLR camera and a smartphone from three different research fields. The average size of the images is 5 MB. There are eight categories of disease,

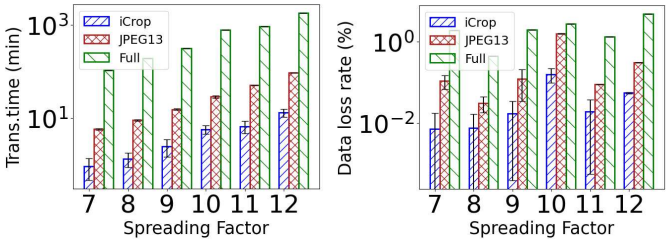
including bacterial blight, cercospora leaf blight, downey mildew, frogeye leaf spot, soybean rust, target spot, potassium deficiency and healthy. The images were taken from multiple locations in the field including within the canopy, on the ground, and within a controlled lab environment. Furthermore, we confirmed the data imbalance pointed out in [31]. As a result, we employ data augmentation through horizontal flipping on images representing bacterial blight, to increase the low number of images within that category.

2) Training and Testing: We train three models for each of the considered approaches. Specifically, we train the model for *Full* with the original images in the dataset. Conversely, we trained the model for *JPEG13* using the same images compressed with a JPEG factor of 13. Finally, in the experiments we are adopting a segmentation of 8×8 for iCrop. As a result, we split the original images in 64 segments and train the model of iCrop accordingly. We also transmit $K = 5$ segments from the LoRa Node to the LoRa Edge Server.

For all models, we divide the dataset into a 60-20-20 split, with 60% designated for training, 20% for validation, and the remaining 20% for testing. The splits are performed evenly across categories as not to overfit any disease category. Each model undergoes 50 training epochs, and the batch size is 32. The training process is conducted using stochastic gradient descent as the optimizer, with a learning rate set at 0.0001 and momentum at 0.9.

D. Experiment Results

Experiment I: Impact of LoRa's Spreading Factor. In this set of experiments, we evaluate the performance under different settings of the *spreading factor* (SF). The SF is a pivotal parameter in LoRa, where a higher SF extends the communication range at the expense of a lower data rate. The bandwidth is set at 250KHZ.



(a) (b)

Fig. 11. Experiment I: Impact of Spreading Factor: (a) Transmission time and (b) Data Loss Rate.

Figure 11 (a) illustrates the average transmission time for a single crop leaf image varying SF values from 7 to 12. These times include the retransmission times for packet loss. Note the log-scale in the y-axis. iCrop stands out by achieving a one-minute transmission time with SF of 7, significantly surpassing the performance of JPEG13 and Full. This is due to the effectiveness of the image pre-processing technique through segmentation and compression at the Lora Node. The impact is accentuated with increasing SF, with iCrop providing a transmission time around 8 times and 138 times smaller compared to JPEG13 and Full, respectively. As SF increases, all approaches exhibit a rise in transmission time. This is due to the reduction in data rate, as shown in Table III.

Figure 11 (b) depicts the data loss rate varying the SF. We calculate the loss rate considering the total number of packets lost, including retransmission attempts, versus the number of packets composing the original image being transmitted. Notably, all approaches exhibit a relatively low packet loss rate, consistently below 5%. This can be attributed to the laboratory environment of this set of experiments, where the proximity of the LoRa Node and the LoRa Edge Server results in a strong signal strength, measured with *Received Signal Strength Indicator (RSSI)*, as indicated in Table III. However, iCrop outperforms the other two approaches by achieving an overall lower data loss rate. This is due to the reduced amount of transmitted data, that implies less packets being lost, and thus less retransmission attempts, and less data loss during the retransmissions as well.

Next, we evaluate the performance of the CNN classification model with respect to accuracy, precision, recall, and F-1 scores. Note that, thanks to our transmission protocol, all approaches are able to transmit the images reliably to the LoRa Server. As a result, the performance are not impacted by the spreading factor, nor by other LoRa settings. Table II summarizes the results.

TABLE II
CLASSIFICATION PERFORMANCE

Methods	Accuracy	Precision	Recall	F1
Full	94.34	94.54	94.35	94.34
JPEG13	92.38	92.77	92.28	92.39
iCrop	90.0	90.02	89.92	89.76

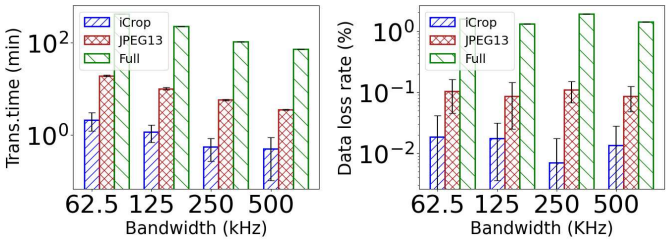
As expected, transmitting the Full image yields the highest accuracy at 94.34%, along with superior precision, recall, and F1-score. Conversely, the accuracy of JPEG13 experiences a decline of approximately 2% across all metrics due to data compression. The overall performance of iCrop is 2% lower than that of JPEG13, attributed to the CNN classification relying on partial information given by the only 5 segments transmitted. Nevertheless, iCrop is able to achieve an accuracy of 90% while requiring a transmission time that is up to 13 times lower.

Note that, since the CNN classification metrics are not impacted by the LoRa settings thanks to our reliable transmission protocol, we omit these results in the next experiments.

Experiment II: Impact of Bandwidth. In this set of experiments, we assess the impact of bandwidth, another important parameter of LoRa. The experiments are also conducted in the laboratory environment, with the SF set at 7 while varying the bandwidth from 62.5KHz to 500KHz. Figure 11 illustrates the averaged transmission time for a single crop leaf image for all three approaches. While the transmission time declines with increasing bandwidth for all approaches, the Full image approach still experiences prolonged transmission times due to

TABLE III
LORA PERFORMANCE WITH DIFFERENT SETTINGS

	Spreading Factor						Bandwidth (KHz)				Distances (m)		
	7	**8**	**9**	**10**	**11**	**12**	**62.5**	**125**	**250**	**500**	**230**	**500**	**800**
RSSI (dBm)	-53.5	-61.5	-66.7	-67.8	-68.1	-53.4	-50.4	-65.8	-53.5	-65.6	-94.6	-136.1	-108.5
Data Rate (byte/s)	715.1	448.5	246.7	100.2	87.7	45.8	200.4	342.5	715.0	940.3	143.7	76.5	86.2



(a) (b)

Fig. 12. Experiment II: (a) Transmission time vs. bandwidth and (b) data loss rate vs. bandwidth.

the necessity of transmitting the entire 5 MB data. Although compression helps alleviate this issue, it still takes up to 10 times longer than iCrop. These results once again underscore the advantages of segment selection performed by iCrop at the LoRa Node, effectively reducing transmission times while maintaining high classification accuracy.

Figure 12 (b) presents the averaged data loss rate across different bandwidths. All approaches maintain a consistent data loss rate with different bandwidths. This is attributed to the stable connection between the LoRa node and LoRa Edge server, with minimal signal interference owing to their proximity. We can see from Table III that the signal strength RSSI is similar across all the bandwidth settings. Nevertheless, iCrop once again results in a reduced data loss rate due to less re-transmission attempts thanks to less data being transmitted.

Experiment III: Impact of distance between LoRa Node and LoRa Edge Server. In this set of experiments, we vary the distance between the LoRa Node and the LoRa Edge Server, with SF 8 and bandwidth 250KHZ. Specifically, the LoRa Edge Server sits close to a window in our laboratory, while the LoRa Node is positioned in three distinct locations at distances of 230m, 500m, and 800m, respectively. Figure 13 shows these locations with respect to the Lora Edge Server. For these experiments, we set the SF at 10 and bandwidth at 250KHz.

We first show the transmission time in Figure 14 (a). Notably, the transmission time does not simply increase with distance. This is due to the environment where the experiments have been carried out, with buildings, trees, and other source of interference affecting the communication. Specifically, at location 1 with a distance of 230m, the LoRa Node has a line-of-sight connectivity to the LoRa Edge Server. This results in a stronger signal strength and a higher data rate, as indicated in Table III, thereby leading to an overall shorter transmission time for all approaches. Conversely, location 2 achieves longer transmission times than location 3, despite the distance being

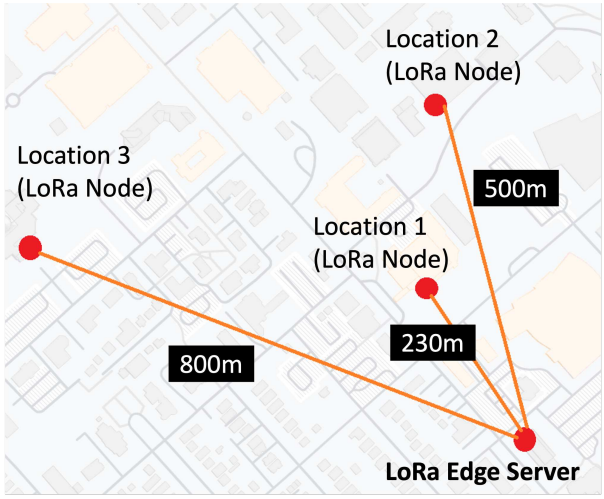
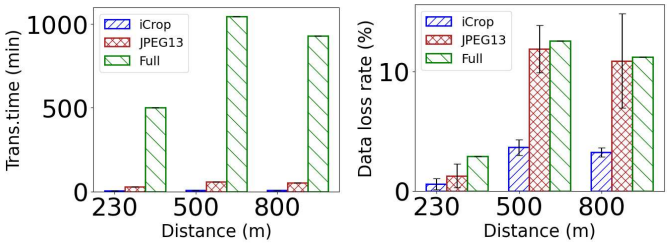


Fig. 13. Map of various LoRa Node locations.



(a) (b)

Fig. 14. Experiment III: (a) Transmission time vs. distance and (b) data loss rate vs. distance.

lower. This is explained by the fact that in location 3, the LoRa node is situated on the 5th floor of a library building, providing a partial line-of-sight connectivity, while in location 2, it is positioned on the 3rd (top) floor of a classroom building, facing obstacles such as buildings and trees in between. This is clear also from Table III, where location 3 has a better RSSI. Similar trends can also be observed in Figure 14 (b) for the data loss rate: location 2 has a slightly higher data loss than location 3, while location 1 has the lowest data loss.

In all the considered scenarios, iCrop significantly outperforms Full and JPEG13. Numerically, iCrop achieves a transmission time of less than 8.5 min, orders of magnitude less than the other approaches. Similarly, the data loss rate is below 3.6%, significantly lower than the comparisons. These findings highlight the benegits of iCrop, particularly in longer-distance scenarios, typical of agricultural settings.

IV. Related Work

In this section, we present the existing research works concerning LoRa technology in the realm of agriculture and the current developments in the field of crop disease detection.

A. LoRa-Enabled Smart Agriculture

LoRa technology offers expansive signal coverage, ranging from 10Km to 15Km in rural line-of-sight scenarios and 3Km to 5Km in urban line-of-sight conditions, coupled with low power consumption [25], [36]. Capitalizing on these advantages, numerous research studies have delved into the suitability and capabilities of LoRa for enhancing agricultural management practices. For instance, the authors of [26] developed a LoRa-based system for monitoring the tomato growth environment within a greenhouse using temperature, humidity, carbon dioxide levels, electrical conductivity, and illuminance data. The authors of [28] propose a similar LoRa-based smart agriculture management and monitoring system to address the communication failure problems in rural farms using sensor networks. Building upon real-time environmental sensor data acquired through LoRaWAN, the authors of [27] introduce a smart decision support system aiming at enhancing crop yield through the optimization of irrigation and fertilizer practices. Similarly, paper [37] proposes a novel approach to integrate LoRa connectivity with the existing Programmable Logic Controllers (PLC) to automate various processes and control farming machines. Furthermore, a recent comprehensive survey [38] delved into the various applications of LoRaWAN technology, exploring its usage in diverse fields beyond agriculture, including smart cities, industrial monitoring, and more.

However, little research has been done in terms of image transmission using LoRa, and its effect on machine learning models. While previous studies [39]–[42] have recommended the utilization of image compression techniques, they still result in significant transmission times as shown in this paper. There remains a lack of work dedicated to exploring the potential and overcoming challenges associated with LoRa-based image transmission, especially in specific agricultural applications, such as crop disease detection and insect identification.

B. Crop Disease Detection

Traditional methods for detecting crop diseases typically involve manual processes, relying on visual inspections and laboratory tests. Unfortunately, this approach is often time-consuming and prone to inaccurate diagnoses. In response to these challenges, machine learning techniques have been employed for automatic disease recognition [9]. These models are normally trained on datasets comprising images of both healthy and diseased plants, with a primary focus on extracting features such as color, shape, and texture.

Deep learning, and particularly CNNs, have gained popularity and exhibited outstanding performance in crop disease detection [10], [43]. For instance, in [11], six deep learning architectures based on CNNs are evaluated for classifying rice diseases using crop leaves. The authors of [31] introduce a novel localization method aimed at improving CNN-based crop disease identification through region of interest segmentation. Paper [13] proposes a transfer learning approach employing CNNs for datasets consisting of original field leaf images. Other works propose the use of unmanned aerial vehicles (UAVs) equipped with multispectral and/or hyperspectral sensors to collect spatial images for crop disease detection [15], [44], [45]. Despite these advancements, the lack of Internet connectivity in rural areas remains a problem for the application of these techniques.

Recent efforts have focused on using On-device AI, where inference models are executed on the end-device. For instance, the authors of [21] introduced Deep Leaf for detecting diseases in coffee plants. Deep Leaf employs model compression and quantization techniques to facilitate the deployment of deep neural networks on end devices. Similarly, in [22], TinyML was proposed for olive fruit variety classification at the edge controller. Despite advancements in on-device AI, notable challenges persist, including memory restrictions for image data processing and accuracy compromises with hard samples. Consequently, the images collected in the field, manually or automatically, need to be transmitted to a server in order to execute complex deep learning models. The lack of cellular coverage, paired with the large number and size of the collected images, constitute a significant bottleneck in adopting these technologies for crop disease classification. In our work, we tackle these challenges by exploring the potential of a low-cost and long-range LoRa-based network through image preprocessing and segment selection.

V. Conclusion

In this paper, we introduce iCrop, an autonomous end-to-end crop disease detection framework that leverages LoRa technology and employs CNN-based deep learning models. iCrop addresses challenges of limited internet coverage in remote rural farms, significantly reducing the on-site monitoring efforts and deployment costs, all while maintaining a high accuracy in crop disease classification. Moreover, the experimental results conducted with a real LoRa testbed demonstrate the superior performance of iCrop compared to existing approaches.

Acknowledgement

We would like to thank the ICCCN 2024 Organizing Committee for the invited paper. This work is partially supported by the NSF Awards CPS # 2133407 and SCC # 1952045.

References

[1] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, "The global burden of pathogens and pests on major food crops," *Nature ecology & evolution*, vol. 3, no. 3, pp. 430–439, 2019.

[2] FAO, *The State of Food and Agriculture 2023. Revealing the true cost of food to transform agrifood systems.* Food & Agriculture Organization., 2023.

[3] E. Casella, M. C. Cantor, M. M. W. Setser, S. Silvestri, and J. H. Costa, "A machine learning and optimization framework for the early diagnosis of bovine respiratory disease," *IEEE Access*, 2023.

[4] E. Casella, M. C. Cantor, S. Silvestri, D. L. Renaud, and J. H. Costa, "Cost-aware inference of bovine respiratory disease in calves using precision livestock technology," in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2022, pp. 109–116.

[5] M. C. Cantor, E. Casella, S. Silvestri, D. L. Renaud, and J. H. Costa, "Using machine learning and behavioral patterns observed by automated feeders and accelerometers for the early indication of clinical bovine respiratory disease status in preweaned dairy calves," *Frontiers in Animal Science*, vol. 3, p. 852359, 2022.

[6] C. Yang, "Remote sensing and precision agriculture technologies for crop disease detection and management with a practical application example," *Engineering*, vol. 6, no. 5, pp. 528–532, 2020.

[7] V. Singh, N. Sharma, and S. Singh, "A review of imaging techniques for plant disease detection," *Artificial Intelligence in Agriculture*, vol. 4, pp. 229–242, 2020.

[8] M. Nagaraju and P. Chawla, "Systematic review of deep learning techniques in plant disease detection," *International journal of system assurance engineering and management*, vol. 11, pp. 547–560, 2020.

[9] J. A. Wani, S. Sharma, M. Muzamil, S. Ahmed, S. Sharma, and S. Singh, "Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges," *Archives of Computational methods in Engineering*, vol. 29, no. 1, pp. 641–677, 2022.

[10] D. S. Joseph, P. M. Pawar, and R. Pramanik, "Intelligent plant disease diagnosis using convolutional neural network: a review," *Multimedia Tools and Applications*, vol. 82, no. 14, pp. 21 415–21 481, 2023.

[11] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of cnn-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, 2023.

[12] D. Hughes, M. Salathé *et al.*, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.

[13] Y. Kurmi, P. Saxena, B. S. Kirar, S. Gangwar, V. Chaurasia, and A. Goel, "Deep cnn model for crops' diseases detection using leaf images," *Multidimensional Systems and Signal Processing*, vol. 33, no. 3, pp. 981–1000, 2022.

[14] R. Manish, Z. An, A. Habib, M. R. Tuinstra, and D. J. Cappelleri, "Agbug: Agricultural robotic platform for in-row and under canopy crop monitoring and assessment," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2021.

[15] L. Kouadio, M. El Jarroudi, Z. Belabess, S.-E. Laasli, M. Z. K. Roni, I. D. I. Amine, N. Mokhtari, F. Mokrini, J. Junk, and R. Lahlali, "A review on uav-based applications for plant disease detection and monitoring," *Remote Sensing*, vol. 15, no. 17, 2023.

[16] X. Tao, E. Damron, and S. Silvestri, "High-precision crop monitoring through uav-aided sensor data collection," in *IEEE International Conference on Communications (ICC)*. IEEE, 2024.

[17] K. Hazra, V. K. Shah, S. Silvestri, V. Aggarwal, S. K. Das, S. Nandi, and S. Saha, "Designing efficient communication infrastructure in post-disaster situations with limited availability of network resources," *Computer Communications*, vol. 164, pp. 54–68, 2020.

[18] K. LoPiccalo, "Impact of broadband penetration on us farm productivity," *Available at SSRN 3790850*, 2021.

[19] V. K. Shah, S. Bhattacharjee, S. Silvestri, and S. K. Das, "Designing sustainable smart connected communities using dynamic spectrum access via band selection," in *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*, 2017, pp. 1–10.

[20] F. C. Commission *et al.*, "Broadband deployment report (2020)," 2020.

[21] F. De Vita, G. Nocera, D. Bruneo, V. Tomaselli, D. Giacalone, and S. K. Das, "Quantitative analysis of deep leaf: a plant disease detector on the smart edge," in *IEEE International Conference on Smart Computing*. IEEE, 2020.

[22] A. M. Hayajneh, S. Batayneh, E. Alzoubi, and M. Alwedyan, "Tinyml olive fruit variety classification by means of convolutional neural networks on iot edge devices," *AgriEngineering*, vol. 5, no. 4, pp. 2266–2283, 2023.

[23] S. Dhar, J. Guo, J. Liu, S. Tripathi, U. Kurup, and M. Shah, "A survey of on-device machine learning: An algorithms and learning theory perspective," *ACM Transactions on Internet of Things*, vol. 2, no. 3, pp. 1–49, 2021.

[24] S. Devalal and A. Karthikeyan, "Lora technology-an overview," in *international conference on electronics, communication and aerospace technology*. IEEE, 2018, pp. 284–290.

[25] M. A. Almuhaya, W. A. Jabbar, N. Sulaiman, and S. Abdulmalek, "A survey on lorawan technology: Recent trends, opportunities, simulation tools and future directions," *Electronics*, vol. 11, no. 1, p. 164, 2022.

[26] R. K. Singh, M. Aernouts, M. De Meyer, M. Weyn, and R. Berkvens, "Leveraging lorawan technology for precision agriculture in greenhouses," *Sensors*, vol. 20, no. 7, p. 1827, 2020.

[27] J. Arshad, M. Aziz, A. A. Al-Huqail, M. H. u. Zaman, M. Husnain, A. U. Rehman, and M. Shafiq, "Implementation of a lorawan based smart agriculture decision support system for optimum crop yield," *Sustainability*, vol. 14, no. 2, p. 827, 2022.

[28] S. Suji Prasad, M. Thangatamilan, M. Suresh, H. Panchal, C. A. Rajan, C. Sagana, B. Gunapriya, A. Sharma, T. Panchal, and K. K. Sadasivuni, "An efficient lora-based smart agriculture management and monitoring system using wireless sensor networks," *International Journal of Ambient Energy*, vol. 43, no. 1, 2022.

[29] N. Ahmed, F. Esposito, O. Okafor, and N. Shakoor, "SoftFarmNet: reconfigurable Wi-Fi HaLow networks for precision agriculture," in *IEEE International Conference on Cloud Networking (CloudNet)*, 2023.

[30] N. Bevers, E. J. Sikora, and N. B. Hardy, "Pictures of diseased soybean leaves by category captured in field and with controlled backgrounds: Auburn soybean disease image dataset (asdid)," *Dryad*, 2022.

[31] ——, "Soybean disease identification using original field images and transfer learning with convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 203, p. 107449, 2022.

[32] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[33] T. Gao, H. Emadi, H. Saha, J. Zhang, A. Lofquist, A. Singh, B. Ganapathysubramanian, S. Sarkar, A. K. Singh, and S. Bhattacharya, "A novel multirobot system for plant phenotyping," *Robotics*, vol. 7, no. 4, p. 61, 2018.

[34] "Raspberry pi 3," https://www.raspberrypi.com/products/raspberry-pi-3-model-b/, 2024, accessed: 01/26/2024.

[35] "Wifi lora 32 v2," https://resource.heltec.cn/download/Manual%20Old/WiFi%20Lora32Manual.pdf, 2024, accessed: 01/26/2024.

[36] X. Tao and S. Silvestri, "Integrating uav and lorawan in wsn for intelligent monitoring in large-scale rural farms," in *IEEE international conference on pervasive computing and communications workshops and other affiliated events (PerCom Workshops)*. IEEE, 2023, pp. 166–167.

[37] M. Saban, M. Bekkour, I. Amdaouch, J. El Gueri, B. Ait Ahmed, M. Z. Chaari, J. Ruiz-Alzola, A. Rosado-Muñoz, and O. Aghzout, "A smart agricultural system based on plc and a cloud computing web application using lora and lorawan," *Sensors*, vol. 23, no. 5, p. 2725, 2023.

[38] V. Bonilla, B. Campoverde, and S. G. Yoo, "A systematic literature review of lorawan: Sensors and applications," *Sensors*, vol. 23, no. 20, p. 8440, 2023.

[39] C. Pham, "Low-cost, low-power and long-range image sensor for visual surveillance," in *Workshop on Experiences in the Design and Implementation of Smart Objects*, 2016, pp. 35–40.

[40] R. Kirichek, V.-D. Pham, A. Kolechkin, M. Al-Bahri, and A. Paramonov, "Transfer of multimedia data via lora," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Springer, 2017, pp. 708–720.

[41] J. Y. Zhang, B. L. Yeung, J. C. Wong, R. C. Cheung, and A. H. Lam, "Lorawan-based camera with (cira) compression and image recovery algorithm," in *IEEE World Forum on Internet of Things (WF-IoT)*, 2021, pp. 136–141.

[42] Z. Zinonos, S. Gkelios, A. F. Khalifeh, D. G. Hadjimitsis, Y. S. Boutalis, and S. A. Chatzichristofis, "Grape leaf diseases identification system using convolutional neural networks and lora technology," *IEEE Access*, vol. 10, pp. 122–133, 2022.

[43] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network applied to plant leaf disease classification," *Agriculture*, vol. 11, no. 8, p. 707, 2021.

[44] A. Guo, W. Huang, Y. Dong, H. Ye, H. Ma, B. Liu, W. Wu, Y. Ren, C. Ruan, and Y. Geng, "Wheat yellow rust detection using uav-based hyperspectral technology," *Remote Sensing*, vol. 13, no. 1, p. 123, 2021.

[45] P. Ahmadi, S. Mansor, B. Farjad, and E. Ghaderpour, "Unmanned aerial vehicle (uav)-based remote sensing for early-stage detection of ganoderma," *Remote Sensing*, vol. 14, no. 5, p. 1239, 2022.