

# A Deep Learning and Social IoT Approach for Plants Disease Prediction Toward a Sustainable Agriculture

Giovanni Delnevo<sup>1</sup>, Roberto Girau<sup>1</sup>, Chiara Ceccarini<sup>1</sup>, and Catia Prandi<sup>2</sup>, *Associate Member, IEEE*

**Abstract**—As the world becomes increasingly interconnected, emerging and innovative sensing technologies are shaping the future of agriculture, with a special focus on sustainability-related issues. In this context, we envision the possibility to exploit Social Internet of Things for sensing of environmental conditions (solar radiation, humidity, air temperature, and soil moisture) and communications, deep learning for plant disease detection, and crowdsourcing for images collection and classification, engaging farmers and community garden owners and experts. Through, data fusion and deep learning, the designed system can exploit the collected data and predict when a plant would (or not) get a disease, with a specific degree of precision, with the final purpose to render agriculture more sustainable. We here present the architecture, the deep learning model, and the responsive Web app. Finally, some experimental evaluations and usability/engagement tests are reported and discussed, together with final remarks, limitations, and future work.

**Index Terms**—Deep learning, plant disease detection, plant disease prediction, Social Internet of Things (SIoT).

## I. INTRODUCTION

WITH the advancement in innovative communication technologies and the rapid growth of smart sensors, the Internet of Things (IoT) has emerged as a new computing paradigm [1]. Thanks to its pervasiveness, IoT is permeating our daily lives, showing the potential to impact several domains, ranging from personal to industrial ecosystems, as enablers for responsible digital transformation [2].

One of the critical domains where the IoT can have a positive impact is sustainability, as reported by the World Economic Forum, in its document titled “IoT guidelines for sustainability” [3]. The report claims that “84% of IoT deployments [considering 2017] are currently addressing, or have the potential to address, the sustainable development goals (SDGs) as defined by the United Nations” and it continues, assessing that the reason why the IoT could become a game-changer for sustainability lies in its technology [3]. Along this line of

thinking, the synergies between IoT solutions and SDGs have been investigated in other studies and reports (e.g., [4]).

Smart cities, smart energy, connected industries, connected health, and smart agriculture are just a few of the sectors where the IoT may provide substantial benefits in terms of sustainability [4]. Focusing on the latter one, several studies are presenting IoT-enabled strategies in the area of smart agriculture [5]–[7]. Nonetheless, limited are the studies where the IoT solutions for smart agriculture are analyzed under a sustainability lens [8]. In this context, we envision the opportunity to exploit IoT and emerging technologies and paradigms to promote sustainable agriculture. In particular, we are interested in investigating the possibility to employ the IoT not only for plants diseases detection, but also prediction, with the final purpose to render agriculture more sustainable, safe and resilient, avoiding expensive use of pesticides in crop protection, and a sustainable pest management [9].

Inspired by previous studies concerning IoT and Social IoT (SIoT) (e.g., [10] and [11]), machine learning in the area of smart agriculture and plant disease diagnosis (e.g., [12] and [13]), crowdsensing and crowdsourcing (e.g., [14]), we investigated a strategy to deploy a sensors infrastructure to collect data and use such data, together with crowdsourced photos, to predict (and control) the plant diseases. We seek the opportunity to provide farmers and community garden owners with an SIoT infrastructure and a responsive Web app for plant diseases prediction (and control), toward sustainable agriculture. In doing that, we designed and developed a system, called *FruGar*, exploiting data fusion of environmental data collected via an SIoT and crowdsourced photos, and automatic image recognition systems and disease prediction based on deep learning. We validated our approach using a small data set about coffee leaf rust. To the best of our knowledge, this is the first study investigating data fusion of environmental data and crowdsourced photos for plants disease prediction employing deep learning.

The remainder of this article is organized as follows. First, Section II details previous studies in the area of 1) machine learning strategies that have been used for plant disease diagnostics; 2) the SIoT paradigm, focusing, in particular, on 3) Lysis, a cloud SIoT architecture. Then, we describe the overall system, describing the system architecture and the micro engines (MEs), while Section IV presents the architecture and the features of the responsive Web application. Some preliminary experimental evaluations are presented in

Manuscript received April 9, 2021; revised June 22, 2021; accepted July 12, 2021. Date of publication July 15, 2021; date of current version May 9, 2022. (Corresponding author: Roberto Girau.)

Giovanni Delnevo, Roberto Girau, and Chiara Ceccarini are with the Department of Computer Science and Engineering, University of Bologna, 40126 Bologna, Italy (e-mail: chiara.ceccarini6@unibo.it; giovanni.delnevo2@unibo.it; roberto.girau@unibo.it).

Catia Prandi is with the Department of Computer Science and Engineering, University of Bologna, 40126 Bologna, Italy, and also with ITI, LARSyS, 9020-105 Funchal, Portugal (e-mail: catia.prandi2@unibo.it).

Digital Object Identifier 10.1109/JIOT.2021.3097379

Section V. Finally, Section VI concludes this article with final remarks, limitations, and strategies for future work.

## II. RELATED WORK

This section discusses some researches related to our study with a particular focus on plant disease detection approaches, the SIoT paradigm, and the Lysis platform, the SIoT platform on which this work relies.

### A. Plant Disease Detection

Deep learning algorithms represent the state-of-the-art for plant disease detection. However, in order to reach a high-level performance of accuracy, they need to be trained on a huge amount of data. Most researches concerning plant disease detection are based on the PlantVillage data set [15]. It is the largest open-access repository of crop images since it contains 54 307-JPEG images. They include leaves of healthy and diseased plants, relative to fourteen different crop species. The diseases were confirmed by expert plant pathologists based on standard phenotyping approaches. The peculiarity of this data set is that the pictures have been taken removing the leaves from the plant and placing it on a grey background, even if this negatively influences the performance of models in real scenarios [16].

This has led to the publication of new data sets which have tried to compensate for this limitation, providing images of leaves in different real-life situations of cultivation fields. Among them, we can cite PlantDoc [17], RoCoLe [18], and the one proposed by Chouhan *et al.* [19]. The most similar is undoubtedly PlantDoc [17], since it covers 13 of the 14 species included in the Plant Village data set (the only missing is Orange) but it consists of only 2598 images. RoCoLe [18] is comprised of 1560 high-resolution images, taken under different environmental conditions, and classified by professionals but they are only relative to coffee leaves. Furthermore, the data set presented in [19] contains 4503 sample images of leaves during different stages of their life cycle. The plants considered are all different from the ones of the Plant Village data set. Finally, there are also a lot of small data sets focused on specific plants [20], [21]. As highlighted, none of these data sets are truly comparable with the Plant Village one, for various reasons ranging from the number of images to the diseases and plants represented. Anyway, all the public data sets available do not provide any additional information other than the images.

Employing these data sets, different studies evaluated several deep learning algorithms [22]. They focused on the most common convolutional neural network (CNN) architectures [23], the impact of transfer learning on them [24], and their interpretability [25]. Unlike the previously mentioned works, Zhao *et al.* [26], instead, employed data fusion to merge images of leaves with other contextual information, such as the season, temperature, and humidity. They designed a multicon-text fusion network, that essentially has two parallel inputs. The first one, that analyzes the images, exploits a CNN while the second, which takes as input the contextual information, simply employs some fully connected layers.

All the systems presented in the literature deal with detecting the current disease of the plant. None of these, however, allow you to predict a future disease of the plant when it is still healthy so that measures can be taken to prevent it.

### B. Social Internet of Things

Recently, several studies have looked at the problems of managing and effectively using large numbers of heterogeneous devices, and have found a solution in the use of social networking principles and technologies. The guiding motivation is that a social-oriented approach is intended to aid in the discovery, collection, and composition of resources and knowledge offered by distributed objects and networks [27].

In [10], the definition of the SIoT has been formalized, and it is intended to be a social network in which each node is an entity capable of forming social relationships with other things on its own, according to the rules set by the owner. The SIoT relies on certain key relationship types.

- 1) *Ownership Object Relationship*: It is established between objects belonging to the same owner.
- 2) *Co-Location Object Relationship (CLOR)*: It is established between fixed devices in the same location.
- 3) *Parental Object Relationship*: It is achieved by connecting objects from the same model, vendor, and production batch.
- 4) *Co-Work Object Relationship*: It is built between objects which meet in the workplace of their owners.
- 5) *Social Object Relationship (SOR)*: It is formed as a result of repeated encounters between objects, such as those that can occur between smartphones of students in the same class.

These relationships are generated and modified based on the characteristics of the objects (such as object type, computing capacity, accessibility capability, and brand) and their operation (frequency of meeting other objects, mainly). Recently, some work has studied how the SIoT can be used to monitor certain types of smartphone activities to minimize users' concerns about accessibility and tracking in smart social spaces [28]. The resulting object social activity can also be used for the management of trust in object-to-object communications [29], [30].

### C. Lysis Implementation of SIoT

The scenario of a community garden consists of a distributed system of heterogeneous devices, both fixed (sensors, valves, etc.) and mobile, such as the smartphones of the users-farmers. IoT applications built on these devices may have to analyze a large amount of information generated by the cooperation between the devices. In such a scenario, we foresee each device being linked to its virtual counterpart in the cloud or at the edge through virtualization technologies. The solution we proposed in this article relies on the cloud SIoT architecture, named Lysis [11], that foresees a four-level structure as described below.

The bottom layer is populated by real-world objects. At this layer, physical devices directly access the platform via

direct links to the Internet, while other objects (more resource-constrained) need to rely on gateways for the Internet connection, allowing them to send data to and receive commands from the level above. Physical devices and gateways are able to perform basic tasks, such as secure communication with the respective virtual counterparts, as well as management and presentation of data coming from sensors. The above layer is the virtualization layer that directly interacts with the real world and is composed of social virtual objects (SVOs). The virtualization functionalities are common to most IoT cloud-based platforms to address most of the issues related to the low level of resources the IoT objects are equipped with [31]. In Lysis, these tasks are enhanced by the social skills of the agent executing the virtual object, allowing it to create and maintain friendships with other SVOs independently of the user. Accordingly, a network of social digital counterparts of the physical devices is created and available at this layer. This can be used to look for objects and knowledge they generate, assess trust levels, and form communities to promote cooperation. The Aggregation layer is responsible for composing several SVOs into entities with extended capabilities, called MEs; the ME is the entity that implements part of the application logic performed at the upper layer. In each ME, the output for a request coming from an application can be reused to serve requests from different applications that require the same information or service to save bandwidth and CPU. Finally, user-oriented macro services are provided at the application layer.

### III. FRUGAR SYSTEM

In this section, we present the overall system architecture, describing the different MEs (the collector, the detector, and the model builder) and how they interact to build new data sets.

#### A. System Architecture

Fig. 1 shows the framework of the FruGar system, which is completely based on the Lysis SIoT architecture presented in Section II-C. The main components are as follows.

- 1) *The Smartphone of the FruGar User*: It has the application to check the plant health status from pictures and from the garden sensors. It includes the Lysis drivers needed to communicate with its virtual counterpart and implement all the features of the SIoT.
- 2) *The Garden Sensors*: Smart and modern cultivation includes sensors for soil monitoring (soil moisture, nutrient concentration, etc.) and sensors for atmospheric monitoring (solar radiation, humidity, and air temperature, etc.). These sensors have access to the Internet and are able to communicate with the virtualization layer.
- 3) *SVOs*: The SM SVO and GS SVO are the virtual counterpart of the smartphones and garden sensors, respectively. These SVOs are created and managed by the virtualization layer of Lysis. By means of the SVOs, Lysis allows the communication between smartphones and garden sensors.

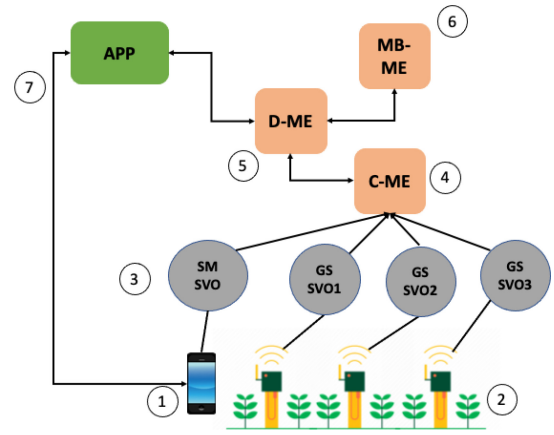


Fig. 1. FruGar system's architecture.

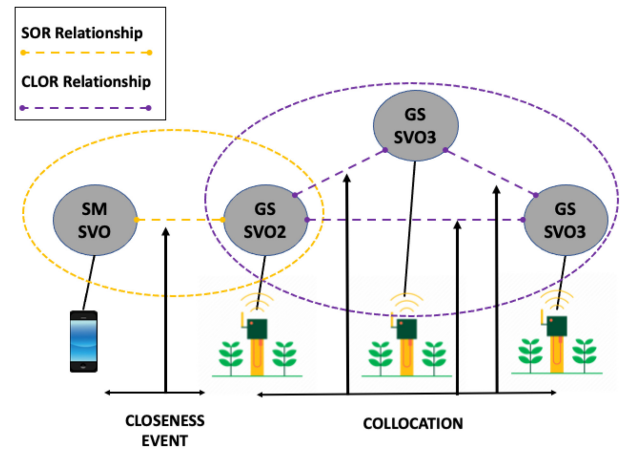


Fig. 2. Building the social graph.

- 4) *The Collecting ME (C-ME)*: An ME in charge to collect data coming from the garden sensors which will be used to build the data set for the classifier. The labeling of the classifier data set is implemented using voluntary feedback provided by people experts in recognizing plant disease.
- 5) *The Detector ME (D-ME)*: It is the classifier that receives the plant picture from the smartphone and the sensors time series from the C-ME.
- 6) *The Model Builder ME (MB-ME)*: The data set is stored on the MB-ME which executes the training process and provides the model to the D-ME.
- 7) *The Mobile App (APP)*: It is a mobile application that allows mobile users to read their garden sensor data, to check the plant's health by taking some pictures, and to give feedback for plant pictures not yet labeled in the classifier data set.

When users logged into the Web app on their smartphones, an SM-SVO is created for the smartphone at the Lysis virtualization layer. The SM-SVO (like all SVOs) provides the socialization features foreseen in the Lysis architecture and necessary to create social relationships, as shown in Fig. 2.

The user's smartphone is brought into the garden where the socialization algorithms exploit closeness detection to establish

the relationships between the SM SVO and the GS SVOs of the nearby sensors. The MAC addresses of the access points are generally used as a point of [32] and [33]. In a similar way, once installed in the garden, the sensors' SVOs create a CLOR-type relationship since they are fixed devices in the same place.

### B. Collector Micro Engine

The detection of a crop disease requires, in addition to the photo taken from the user's smartphone, all the data of the sensors belonging to other users and monitoring various environmental parameters in the garden. To this, the C-ME performs three operations: 1) it tunes the sampling frequency of each involved sensor accordingly to the requirements set by the experts in order to build the right data set; 2) it sends a request for sensors relating to cultivation; and 3) it requests and aggregates the data measured in the previous days from the sensors found in the search process. The sampling frequency setting is possible thanks to the HW abstraction guaranteed by the SVOs and which allows us to use the sensors in the same way with simple and uniform APIs. The search process begins with a query to the SM-SVO of the user acting as root in the social graph. The query can provide some description tags in order to fully describe the type of sensors it is looking for, the place in which they are located, the context, and what kind of social relationships they are connected to each other. In the best case, the smartphone has a direct relationship (one hop of separation) with all the sensors of the crop and it will be a SOR relationship as the user (with her smartphone) frequently goes to the garden. In the worst case, the smartphone has a relationship only with some of the sensors, but all the sensors in the garden have a CLOR relationship with each other as they are all located in the same place (the garden). In the latter case, the search is spread over two hops of separation. The query is passed in JSON format and includes the following.

- 1) The key to verify access permissions (if different from "public").
- 2) The maximum number of resources requested (limit).
- 3) The depth of the search in the social graph (hop).
- 4) The geographical area and a description of the resources in text format in the description parameter.
- 5) The type of relationship that can be exploited the SVO search process, specified in the relationship field.

The output of the search process will be a list of GS SVOs' resources with their access keys.

### C. Detector Micro Engine

The D-ME has two main tasks. The former consists of detecting the current disease of the plant using a simple CNN, as described in [34]. Once a disease is detected, all the data relative to such a plant are aggregated and extracted by the Collector ME to be added to the data set for the disease prediction, which is the latter task of the engine. In fact, once there are data available to a specific disease of a plant, the model for predicting a future disease of the plant is trained. Its architecture is depicted in Fig. 3. It takes two parallel inputs. The first one consists of all the data collected by the

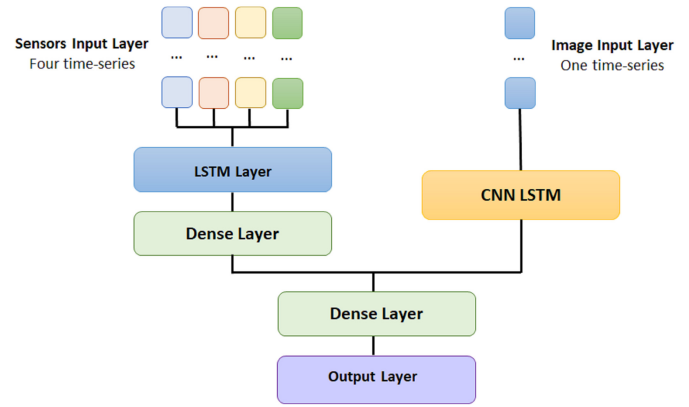


Fig. 3. Proposed deep neural network to predict future plant disease.

different sensors. There is a time series for each type of sensor (in example four). They passed through a long short-term memory (LSTM) layer [35] and a fully-connected one. The second input is for the image series. In this case, we exploit a CNN LSTM [36] with the aim of taking advantage of the CNN to extract features from the images and employing the LSTM to process the sequences of features extracted. Then, the two outputs of these two branches are concatenated together. A fully connected layer deals with making high-level reasoning on the combination of the extracted features, both images and sensors data series. Finally, there is the output layer with softmax as the activation function which gives an indication of the state of the health of the plant together with the relative probabilities.

### D. Model Builder Micro Engine and Data Set Definition

The assessment of possible plant disease may not be possible due to the lack of a pre-loaded decision model. In this case, the proposed system is able to build a data set starting from the history of the measurements made by the sensors in the crop of interest. By exploiting the C-ME, it is possible to collect and aggregate the data from the social network. Essentially, they are a list of the data collected by the sensors together with the evaluation of the human expert who indicates the moment the plant gets sick. An example is reported in Table I, where the sensed data are: 1) temperature; 2) humidity; 3) pressure; and 4) soil moisture. They are just an example and some or all of them could be replaced with other sensors, such as ph or illuminance. In the Disease column, depending on the plant and the disease, the values could be only healthy and diseased or the experts could identify different stages of the disease (such as early and late blight for tomatoes).

In addition to these data, there also images of the plants. Instead of being collected using sensors, they are taken using the Web application directly from the users. For this reason, the system does not require these to be collected at regular time intervals but it simply saves also the current time. Since in a real-world end-use scenario of the FruGar system, pictures can be taken in different settings and environmental conditions, it is important to collect data sets where the photos are



TABLE I  
EXAMPLE OF RAW DATA COLLECTED BY FOUR POSSIBLE SENSORS

Temperature	Humidity	Pression	Soil Moisture	Disease
t1	h1	p1	sm1	d1
t2	h2	p2	sm2	d2
t3	h3	p3	sm3	d3
t4	h4	p4	sm4	d4
t5	h5	p5	sm5	d5
t6	h6	p6	sm6	d6
t7	h7	p7	sm7	d7
t8	h8	p8	sm8	d8
t9	h9	p9	sm9	d9
t10	h10	p10	sm10	d10

taken in different positions and conditions. Otherwise, a significant decline in system performance could be observed as highlighted in [37]. For these reasons, the system allows to collect also new images.

Starting from these raw data, a data set can be generated according to four main parameters, which can be set according to the plant and disease.

- 1) **F**: The frequency with which we consider the data collected, that depends on the sensors sampling rate.
- 2) **N**: The number of instants used to build the input.
- 3) **NI**: The number of last images used as input.
- 4) **T**: How far we predict the possible disease of the plant?

$F$  can be an integer greater than or equal to 1. If  $F$  is equal to 1, we employ the same sampling rate of the sensors. Hence, we consider all the rows in the Table. Instead, if for example, it is equal to 2, we consider only half of the raw data, the ones collected in an instant of time with an odd index.

Then,  $N$  is the number of rows (without the Disease column) using as input. Considering  $F$  equal to 1,  $N$  equal to 4 implies that the input of the final data set will consist of four rows of the raw data. Hence, the first input sample will be composed of rows 1 to 4, the second of rows from 2 to 5, and so on.

Finally,  $T$  is the number of lines, following the last input line, in which we select the disease item as output. Considering  $F$  equal to 1 and  $N$  equal to 4, if we set the value of  $T$  to 6, the output of the first sample will be *d10*.

Once the data are collected, an evaluation of the best combination of these three parameters has to be carried out. Obviously, it will depend on the plant and the disease considered. For this reason, it might be useful a discussion with domain experts also at this stage. As  $N$  varies, not only the length of the data series that the model will have to manage will change but also the minimum time required to be able to make a prediction. Even more interesting, it will be finding the best value of  $T$ . A too small value makes it impossible to take timely action on the plant to prevent the disease, while it is impossible to predict the disease too early.

At this point, it is possible to use the generated data set to train a deep learning model. Since the diseases of a plant can be very different from each other and can spread differently according to the type of plant, our choice is to train specific models for each plant and disease.

#### IV. RESPONSIVE WEB APPLICATION

We designed and developed a responsive Web application to provide support to citizens in their activities related

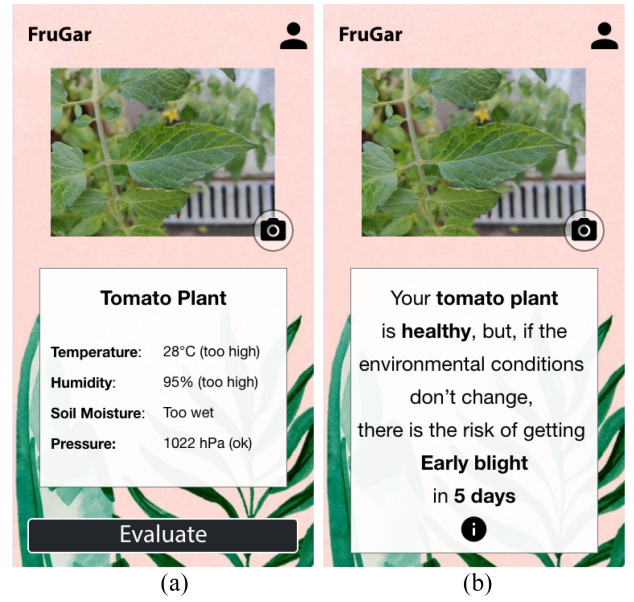


Fig. 4. Screenshots of the user interface of the Web application. (a) Plant's data. (b) Future prediction.

to community gardening and urban farming. Initially, we designed a preliminary version of the system, as presented in [34]. We exploited the smartphone through a frugal innovation lens, taking advantages of the ubiquitous built-in smartphone components (such as the camera) in the context of home and community gardening, to assist casual citizens in gardening activities. We here present an extended and refined version, with the main goal of employing machine learning, crowdsourcing, and SIoT to inform the prediction of future plant diseases, based on historical sensed environmental conditions. In doing that, we considered, as the main target audience, farmers, and community garden owners who are interested in exploiting low-cost solutions for sustainable agriculture through a sustainable disease prediction/control system.

Presenting the application in detail, the back-end is built in Node.js.<sup>1</sup> It provides some APIs to get information about users' accounts and, most importantly, it communicates directly with our D-ME, through the dedicated APIs, to get the status of the user's plants. Concerning the front-end of our prototype, we used Vue.js,<sup>2</sup> a JavaScript framework to build the Web application, which needs to be responsive, as it will be enjoyed mainly on smartphones.

In Fig. 5, the sequence diagram presenting the workflow of the app core functions, i.e., detection and prediction, is depicted. Accordingly, to check the health or disease of the plant, the application requires the user to open the application and selected the plant of interest. The interface in Fig. 4(a) will appear with all of the latest data taken from the sensors and some textual information that indicate if temperature, humidity, soil moisture, and pressure are good or harmful to the plant. When the user clicks on the evaluate button at the bottom of the interface, s/he has to select the type of the plant

<sup>1</sup><https://nodejs.org/it/>

<sup>2</sup><https://vuejs.org/>

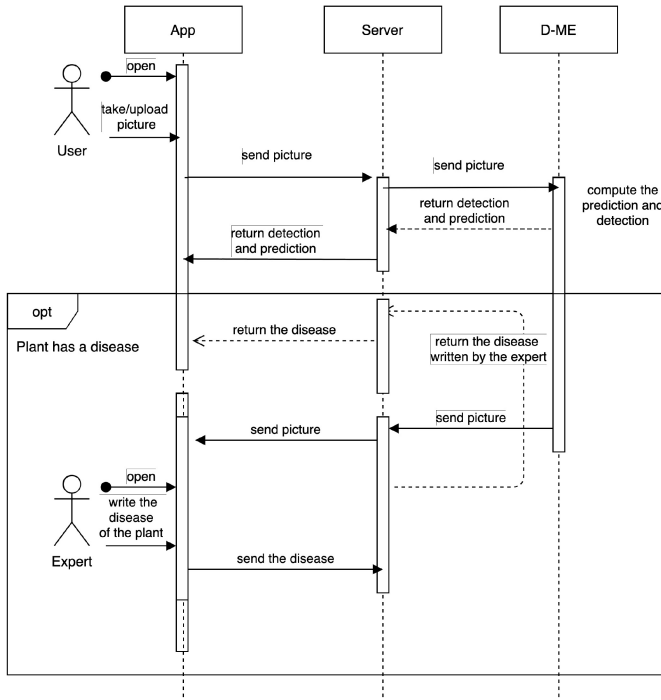


Fig. 5. Sequence diagram presenting the App workflow.

(from a drop-down menu), and to provide a photo. As soon as the user has uploaded the photo of the plant, it is sent to our back-end which forwards it to our D-ME. The prediction made by the D-ME, exploiting the deep learning model presented in Section III-C, is then returned to our back-end and displayed in the Web application [as shown in Fig. 4(b)]. The users can now see if the plant is healthy or if there is a chance of developing some disease, with some related information about how to prevent it. When the D-ME detects a diseased plant, such a photo is forwarded to an expert (as depicted at the bottom of Fig. 5) who will evaluate it, confirming or modifying the D-ME prediction.

In addition, each user can contribute to our application by providing images of healthy or diseased plants of new crop species, contributing to the system in a crowdsourcing fashion.

## V. EXPERIMENTAL EVALUATION

Using a system based on the Lysis architecture allows us to isolate sensor data acquiring concerns from data aggregation and analysis problems. In fact, the virtualization layer is significant in that it provides an HW abstraction that allows all sensors to be managed in the same way, with uniform APIs for reading sensor data, setting sample frequencies, and controlling actuators, such as irrigation valves or greenhouse windows. For this, we focused on the components of analysis and prediction. In this Section, we present 1) an experimental evaluation of both plant disease detection and prediction; 2) a preliminary app evaluation, in terms of usability and engagement, involving some users (four community garden owners and two urban farmers).

TABLE II  
ACCURACY OF CNNs ON VALIDATION AND TEST SET

Model	Accuracy	F1-score
DenseNet121	94.51%	94.51%
MobileNet	93.17%	93.14%
MobileNetV2	94.58%	94.58%
NASNetMobile	93.97%	93.98%

### A. D-ME Evaluation

We conducted different experiments for both tasks of the D-ME, the plant disease detection, and prediction. For the disease detection, we evaluated different CNN architectures: DenseNet121, MobileNet, MobileNetV2, and NasNetMobile. With respect to those used in [34], such architectures are much lighter, having far fewer parameters. In this way, they can be used in any node of the system even directly on users' smartphones. They have been trained on the PlantVillage data set, splitting it into 80% for training and 20% for testing. During training, we augmented the data by rotating and flipping images and the 10% of the images were used for validation. We employed the Tensorflow 2.0 framework. All the architectures have been trained from scratch, without using transfer learning, for 100 epochs using Adam with a learning rate of 0.001 and a decay of 0.0005. Table II reports the accuracy and the F1-score of each architecture on the test set.

Given that the data collection process is still in progress, we evaluated the plant prediction capabilities using an already available data set, the Coffee Leaf Rust data set [38], which contains three-month data about coffee plants. Data consists of environmental humidity and temperature, pH, soil moisture, soil temperature, and illuminance and are sampled by the sensors on average seven times a day. In addition to them, there are also pictures of the plants. Since they have been taken by humans, for each sampling of sensors, they are not always available. Each sample has been labeled by a team of biologists, that has indicated the severity of the Coffee Leaf Rust development stage. For the data set definition, we used the following values for the parameters:  $F = 1$ ,  $N = 2$ ,  $NI = 1$ , and we varied  $T$  from 35 to 70. This implies that we use input series composed of two values and we predict the plant disease from five (considering that there are seven samples per day) to ten days later. We employed only one image, given the fact that they were not always available. Depending on the  $T$  parameter, we have a number of samples that varies from 1930 ( $T = 35$ ) to 1790 ( $T = 70$ ). Given the limited amount of samples, we slightly modified the architecture depicted in Fig. 3. With regard to the sensors part, there are six input series of two elements. The LSTM and the Dense layers are composed respectively of 32 and 8 neurons. In the image part, instead, we use a single CNN instead of a CNN LSTM, since we have only one image. The CNN employed has three pairs of Convolutional and Max Pooling layers, followed by Dense, Batch normalization, Dropout, and Dense layers. Finally, the two parallel branches are concatenated and passed as input to a Dense layer of four neurons, which precedes the output layer. The model has been implemented using the Keras framework. We chose Adam as the optimizer with a learning rate of 0.001

TABLE III  
PLANT PREDICTION PERFORMANCE ON THE TEST SET VARYING T

T	Precision	Recall	F1-score
35	0.99	0.99	0.99
42	0.97	0.98	0.97
49	0.97	0.94	0.95
56	0.99	0.87	0.92
63	0.99	0.71	0.83
70	0.92	0.65	0.76

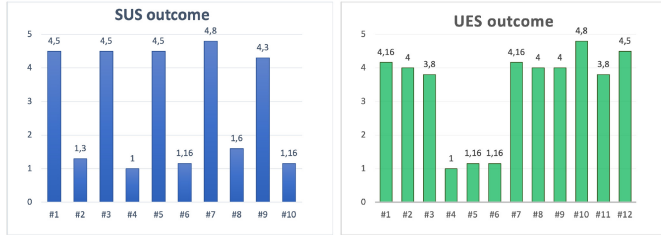


Fig. 6. Average outcome obtained considering each SUS and UES item.

and a decay of 0.0005. The training and test sets are composed respectively of the 80% and the 20% of the examples. As evaluation metrics, we employed precision, recall, and F1-score, since the data set is not perfectly balanced. The results are reported in Table III.

### B. App Evaluation

To evaluate the App, we engaged six users (four community garden owners and two urban farmers; average age: 37, min: 22, max: 54) in a usability session. The session was designed to assess the User Interface usability (employing the System Usability Scale—SUS [39]), and the level of engagement (employing the User Engagement Scale—UES - short form [40]). Participation was voluntary, and all participants had the right to comply with or refuse participation, considering also details about the data storage and analysis (accordingly with European General Data Protection Regulation). We engaged users through snowball sampling. Unfortunately, due to the COVID-19 pandemic, we were able to recruit only six users. Nonetheless, the literature states that six users, under specific conditions, can discover circa 90% of usability issues [41]. During the session, we asked each user to enjoy FruGar and perform some defined tasks (such as taking a picture and verify the plant health status). After this phase, the users answered an online questionnaire comprised of 22 items (ten items from SUS and 12 from UES, short form), using a 5-point Likert scale (from 1 = strongly disagree to 5 = strongly agree). The score obtained analyzing the SUS responses was very high, 90 out of 100 (a score above 68 is considered above average). The average score for each question is presented in Fig. 6, left side. Interesting is to notice that the average score of the question “I would imagine that most people would learn to use this system very quickly.” (item #7 in SUS) obtained a score of 4.8 out of 5.

Considering the level of engagement (Fig. 6, right side), we obtained an average score of 4.075 (out of five, strongly agree), analyzing all the items expressed with a positive tone. Interesting to notice that, the average score of the question

“Using FruGar was worthwhile” (item #10 in UES, short form) got a score of 4.8 out of 5. The collected data confirm the possibility to release FruGar to a large amount of users, to perform on-the-field evaluations.

## VI. CONCLUSION AND FUTURE WORK

In this article, we present an approach that takes advantage of different emerging technologies and paradigms, such as SIoT for sensing environmental conditions, deep learning for plant disease detection, crowdsourcing for engaging citizens, and experts, with the final aim of promoting sustainable agriculture. Our approach has been designed to perform data fusion of environmental sensed data (such as solar radiation, humidity, air temperature, and soil moisture) and plant photos crowdsourced by users, to predict the probability for a plant to get a disease (based on the historical gathered environmental conditions and related collected photos), exploiting deep learning. We also exploit experts to validate the predictions and correct labeling of the new images. Some preliminary experiments are presented, to validate the precision of the deep learning model and to prove the usability and engagement level of the Frugal Web app, engaging six urban farmers/community garden owners.

The most important limitation of this work is surely the limited test of the deep neural network, that has been evaluated only on a small data set about coffee leaves. The reasons are manifold. First of all, as previously mentioned, we have found only one data set similar to those ones collectible using the proposed system. Even if we have started the data collection process, we have to collect a large amount of data to be able to train such a model and this process has been slow down by the global health emergency and the imposed individual restrictions. Moreover, the plants in which we have placed the sensors must get a disease to collect the appropriate data and this has not been possible so far given the Italian climate (spring has just begun). As soon as enough raw data are available, we will train further models and evaluate which is the best combination for the various hyper-parameters presented in this work. Anyway, the preliminary tests on the plant prediction performance presented are encouraging. As well as, we here prove the potential and precision of the implemented deep learning model for plant detection, and, finally, the advantages of using an SIoT architecture.

Moreover, as future work, we would like to deeper investigate the advantages of using an SIoT architecture, exploiting the plants’ proximity, improving the performance and efficiency of the prediction model.

### ACRONYMS

CNN	Convolutional Neural Network
CLOR	Co-location Object Relationship
C-ME	Collecting Micro Engine
D-ME	Detector Micro Engine
GS SVO	Garden Sensor Social Virtual Objects
LSTM	Long Short-Term Memory
ME	Micro Engines
MB-ME	Model Builder Micro Engine

PD	Physical Devices
SIoT	Social Internet of Thing
SM SVO	Smartphone Social Virtual Object
SOR	Social Object Relationship
SVO	Social Virtual Objects
SUS	System Usability Scale
UES	User Engagement Scale.

#### ACKNOWLEDGMENT

The authors thank all the users involved in the evaluation phase. A special thank goes to Rei Beshiri and Manuel Gabrielli for their technical assistance in some preliminary experiments and system implementation.

#### REFERENCES

- [1] C. Chang, S. N. Srirama, and R. Buyya, "Internet of Things (IoT) and new computing paradigms," *Fog and Edge Computing: Principles and Paradigms*, vol. 6. Hoboken, NJ, USA: Wiley, 2019 pp. 1–23.
- [2] A. Aagaard, M. Presser, and T. Andersen, "Applying IoT as a leverage for business model innovation and digital transformation," in *Proc. Global IoT Summit (GloTS)*, Aarhus, Denmark, 2019, pp. 1–5.
- [3] World Economic Forum. (2018). *Internet of Things Guidelines for Sustainability*. Accessed: May 19, 2021. [Online]. Available: <http://www3.weforum.org/docs/IoTGuidelinesforSustainability.pdf>
- [4] A. Renda and M. Laurer, "IoT 4 SDGs—What can the digital transformation and IoT achieve for agenda 2030?" Centre Eur. Policy Stud., Brussels, Belgium, Rep. CEPS Papers 26658, 2020. Accessed: May 19, 2021. [Online]. Available: <https://ideas.repec.org/p/eps/cepswp/26658.html>
- [5] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3758–3773, Oct. 2018.
- [6] T. Ojha, S. Misra, and N. S. Raghuwanshi, "Internet of Things for agricultural applications: The state of the art," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 10973–10997, Jul. 2021.
- [7] N. Ahmed, D. De, and I. Hussain, "Internet of Things (IoT) for smart precision agriculture and farming in rural areas," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4890–4899, Dec. 2018.
- [8] A. Chakrabarty and T. Mudang, "Smart and sustainable agriculture through IoT Interventions: Improvisation, innovation and implementation—An exploratory study," in *IoT and Analytics for Agriculture (Studies in Big Data)*, vol. 63. Singapore: Springer, 2019, pp. 229–240.
- [9] S. Traversari, S. Cacini, A. Galieni, B. Nesi, N. Nicastro, and C. Pane, "Precision agriculture digital technologies for sustainable fungal disease management of ornamental plants," *Sustainability*, vol. 13, no. 7, p. 3707, 2021.
- [10] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social Internet of Things (sIoT)—When social networks meet the Internet of Things: Concept, architecture and network characterization," *Comput. Netw.*, vol. 56, no. 16, pp. 3594–3608, 2012.
- [11] R. Girau, S. Martis, and L. Atzori, "Lysis: A platform for IoT distributed applications over socially connected objects," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 40–51, Feb. 2017.
- [12] W.-L. Chen, Y.-B. Lin, F.-L. Ng, C.-Y. Liu, and Y.-W. Lin, "RiceTalk: Rice blast detection using Internet of Things and artificial intelligence technologies," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1001–1010, Feb. 2020.
- [13] S. H. Lee, H. Goëau, P. Bonnet, and A. Joly, "New perspectives on plant disease characterization based on deep learning," *Comput. Electron. Agr.*, vol. 170, Mar. 2020, Art. no. 105220.
- [14] A. Coletta, N. Bartolini, G. Maselli, A. Kehs, P. McCloskey, and D. P. Hughes, "Optimal deployment in crowd sensing for plant disease diagnosis in developing countries," *IEEE Internet Things J.*, early access, Jun. 15, 2020, doi: [10.1109/IIOT.2020.3002332](https://doi.org/10.1109/IIOT.2020.3002332).
- [15] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, p. 1419, Sep. 2016.
- [16] R. I. Hasan, S. M. Yusuf, and L. Alzubaidi, "Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion," *Plants*, vol. 9, no. 10, p. 1302, 2020.
- [17] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in *Proc. 7th ACM IKDD CoDS 25th COMAD*, 2020, pp. 249–253.
- [18] J. Parraga-Alava, K. Cusme, A. Loor, and E. Santander, "RoCoLe: A robust coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition," *Data Brief*, vol. 25, Aug. 2019, Art. no. 104414.
- [19] S. S. Chouhan, U. P. Singh, A. Kaul, and S. Jain, "A data repository of leaf images: Practice towards plant conservation with plant pathology," in *Proc. 4th Int. Conf. Inf. Syst. Comput. Netw. (ISCON)*, Mathura, India, 2019, pp. 700–707.
- [20] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Brief*, vol. 26, Oct. 2019, Art. no. 104340.
- [21] K. Tian, J. Li, J. Zeng, A. Evans, and L. Zhang, "Segmentation of tomato leaf images based on adaptive clustering number of k-means algorithm," *Comput. Electron. Agr.*, vol. 165, Oct. 2019, Art. no. 104962.
- [22] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosyst. Eng.*, vol. 144, pp. 52–60, Apr. 2016.
- [23] S. S. Pradhan and R. Patil, "Comparison of deep learning approaches for plant disease detection," in *Proc. Int. Conf. Wireless Commun.*, 2020, pp. 575–583.
- [24] K. Kamal, Z. Yin, B. Li, B. Ma, and M. Wu, "Transfer learning for fine-grained crop disease classification based on leaf images," in *Proc. 10th Workshop Hyperspectral Imag. Signal Process. Evol. Remote Sens. (WHISPERS)*, Amsterdam, The Netherlands, 2019, pp. 1–5.
- [25] M. Brahimi, S. Mahmoudi, K. Boukhalfa, and A. Moussaoui, "Deep interpretable architecture for plant diseases classification," in *Proc. Signal Process. Algorithms Archit. Arrangements Appl. (SPA)*, Poznan, Poland, 2019, pp. 111–116.
- [26] Y. Zhao *et al.*, "An effective automatic system deployed in agricultural Internet of Things using multi-context fusion network towards crop disease recognition in the wild," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106128.
- [27] A. M. Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, "The cluster between Internet of Things and social networks: Review and research challenges," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 206–215, Jun. 2014.
- [28] F. Al-Turjman, "5G-enabled devices and smart-spaces in social-IoT: An overview," *Future Gener. Comput. Syst.*, vol. 92, pp. 732–744, Mar. 2019.
- [29] L. Ding, P. Shi, and B. Liu, "The clustering of Internet, Internet of Things and social network," in *Proc. 3rd Int. Symp. Knowl. Acquisition Model.*, Wuhan, China, 2010, pp. 417–420.
- [30] L. Militano, A. Orsino, G. Araniti, M. Nitti, L. Atzori, and A. Iera, "Trust-based and social-aware coalition formation game for multihop data uploading in 5G systems," *Comput. Netw.*, vol. 111, pp. 141–151, Dec. 2016.
- [31] M. Nitti, V. Pilloni, G. Colistra, and L. Atzori, "The virtual object as a major element of the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1228–1240, 2nd Quart., 2016.
- [32] R. Girau, S. Martis, and L. Atzori, "Neighbor discovery algorithms for friendship establishment in the social Internet of Things," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Reston, VA, USA, 2016, pp. 165–170.
- [33] L. Atzori *et al.*, "Smart devices in the social loops: Criteria and algorithms for the creation of the social links," *Future Gener. Comput. Syst.*, vol. 97, pp. 327–339, Aug. 2019.
- [34] C. Ceccarini, G. Delnevo, and C. Prandi, "FruGar: Exploiting deep learning and crowdsourcing for frugal gardening," in *Proc. 1st Workshop Exp. Design Implement. Frugal Smart Objects*, Sep. 2020, pp. 7–11.
- [35] J. Schmidhuber, F. Gers, and D. Eck, "Learning nonregular languages: A comparison of simple recurrent networks and LSTM," *Neural Comput.*, vol. 14, no. 9, pp. 2039–2041, Sep. 2002.
- [36] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, South Brisbane, QLD, Australia, 2015, pp. 4580–4584.
- [37] M. Rocchetti, G. Delnevo, L. Casini, and S. Mirri, "An alternative approach to dimension reduction for Pareto distributed data: A case study," *J. Big Data*, vol. 8, no. 1, pp. 1–23, 2021.
- [38] D. Velasquez *et al.* (2020). *Coffee Leaf Rust Dataset*. [Online]. Available: <https://dx.doi.org/10.21227/pz2n-y676>
- [39] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Int. J. Hum. Comput. Interact.*, vol. 24, no. 6, pp. 574–594, 2008.
- [40] H. L. O'Brien, P. Cairns, and M. Hall, "A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form," *Int. J. Hum. Comput. Stud.*, vol. 112, pp. 28–39, Apr. 2018.
- [41] J. Nielsen and T. K. Landauer, "A mathematical model of the finding of usability problems," in *Proc. INTERACT CHI Conf. Hum. Factors Comput. Syst.*, 1993, pp. 206–213.