agriculture

MDPI

Article

An Integrated IoT Sensor-Camera System toward Leveraging Edge Computing for Smart Greenhouse Mushroom Cultivation

Hoang Hai Nguyen [1], Dae-Yun Shin [1,2], Woo-Sung Jung [3], Tae-Yeol Kim [2] and Dae-Hyun Lee [4,*]

[1] Sejong Rain Co., Ltd., In-House Venture of K-Water, Daejeon 34134, Republic of Korea; hai.nh1812@gmail.com (H.H.N.); pushindy@gmail.com (D.-Y.S.)
[2] Graduate School of Smart Agriculture, Chungnam National University, Daejeon 34134, Republic of Korea; kx3057@naver.com
[3] K-Water Research Institute, Daejeon 34045, Republic of Korea; usjung@kwater.or.kr
[4] Department of Biosystems Machinery Engineering, Chungnam National University, Daejeon 34134, Republic of Korea
* Correspondence: leedh7@cnu.ac.kr

Abstract: Industrial greenhouse mushroom cultivation is currently promising, due to the nutritious and commercial mushroom benefits and its convenience in adapting smart agriculture technologies. Traditional Device-Cloud protocol in smart agriculture wastes network resources when big data from Internet of Things (IoT) devices are directly transmitted to the cloud server without processing, delaying network connection and increasing costs. Edge computing has emerged to bridge these gaps by shifting partial data storage and computation capability from the cloud server to edge devices. However, selecting which tasks can be applied in edge computing depends on user-specific demands, suggesting the necessity to design a suitable Smart Agriculture Information System (SAIS) architecture for single-crop requirements. This study aims to design and implement a cost-saving multilayered SAIS architecture customized for smart greenhouse mushroom cultivation toward leveraging edge computing. A three-layer SAIS adopting the Device-Edge-Cloud protocol, which enables the integration of key environmental parameter data collected from the IoT sensor and RGB images collected from the camera, was tested in this research. Implementation of this designed SAIS architecture with typical examples of mushroom cultivation indicated that low-cost data pre-processing procedures including small-data storage, temporal resampling-based data reduction, and lightweight artificial intelligence (AI)-based data quality control (for anomalous environmental conditions detection) together with real-time AI model deployment (for mushroom detection) are compatible with edge computing. Integrating the Edge Layer as the center of the traditional protocol can significantly save network resources and operational costs by reducing unnecessary data sent from the device to the cloud, while keeping sufficient information.

Keywords: smart agriculture; mushroom; edge computing; farm management information system (FMIS); machine vision; Agricultural IoT

Citation: Nguyen, H.H.; Shin, D.-Y.; Jung, W.-S.; Kim, T.-Y.; Lee, D.-H. An Integrated IoT Sensor-Camera System toward Leveraging Edge Computing for Smart Greenhouse Mushroom Cultivation. *Agriculture* **2024**, *14*, 489. https://doi.org/10.3390/agriculture14030489

Academic Editor: Wei Lu

Received: 13 January 2024
Revised: 25 February 2024
Accepted: 5 March 2024
Published: 18 March 2024

1. Introduction

Nowadays, edible mushrooms are well known as not only a healthy food but also a valuable pharmacy, since they can provide rich nutrients such as protein, minerals, and vitamins as well as increase the human immune system to prevent many diseases, even cancer [1,2]. Therefore, the demand for mushroom products is increasing in the world's food market, especially after the COVID-19 pandemic [3,4]. Although wild mushrooms can be harvested in the natural environment, their high sensitivity to seasonal variations of weather/climate factors leads to unsustainable production [5,6]. Industrialized mushroom cultivation in the greenhouse offers an alternative strategy to natural cultivation to improve both the quantity and quality of mushroom production because the key environmental parameters affecting mushroom growth can be monitored and controlled. However, the traditional cultivation method based on manual monitoring, controlling, and harvesting involves time-consuming and labor-intensive work which can then increase operational costs, demanding timely, automated, cost-effective, and eco-friendly solutions for the mushroom industry [7,8].

Along with the Fourth Industrial Revolution (Industry 4.0) when innovative technologies such as the Internet of Things (IoT), cloud computing, Big Data (BD), and Artificial Intelligence (AI) have been commonly employed in industrial automation, a parallel revolution in agriculture (Agriculture 4.0) boosts automation in farming activities by applying those new technologies to the agriculture sector, so-called smart agriculture/farming (or precision agriculture, digital agriculture, and Agricultural IoT). Although it can contribute to the reduction in labor costs, deploying smart agriculture systems in practice is facing other cost-related problems such as setup costs and running costs [9]. On the one hand, from the users' point of view, especially in some developing countries where most people working in agriculture are smallholder farmers, they have limited access to modern smart farming infrastructure due to the expensive product prices (setup costs). On the other hand, from the perspective of service developers/providers, smart farming system construction and applications, which are commonly implemented in centralized cloud servers can consume massive resources for BD storage, management, analytics, and security, resulting in higher overall cloud service costs (running costs). Cutting those unnecessary costs is therefore pivotal for many business organizations, especially small and medium enterprises (SMEs), to optimize their smart farming systems toward serving smallholder clients with cost-saving infrastructures.

Besides the cloud service costs, traditional smart agriculture systems are often limited by the slow network connection (low bandwidth) between Agricultural IoT devices and cloud servers when all the huge numbers of data collected from the devices are directly sent to the cloud, which then delays the system decisions and hampers real-time applications in smart agriculture (high latency) [10–12]. This is a challenge for mushroom cultivation when its key parameters require strict monitoring within short time intervals [13]. Edge computing (including cloudlets, fog, and mobile edge computing) has recently emerged as a promising solution to bridge this connection gap, thanks to the rapid advancement of information and communication technologies as well as hardware capacity. Specifically, an edge computing node can be regarded as a decentralized local server (edge server) that allows for storing a small number of data and bringing data computation closer to the edge devices [14–16]. By coupling the edge computing paradigm with a classical Device-Cloud protocol, the original BD is effectively managed in an offline process at the edge server before transmission instead of storing and processing all the collected raw data at the cloud server [17]. This can free up the bandwidth for faster connection and for the cloud server load spent on complicated analytics and applications [16,18]. Despite the wide deployment of edge computing in smart homes and smart cities, it is not very commonplace in smart farming, and thus there is still room for leveraging and improving this paradigm to meet the demands of smart agriculture as well as the greenhouse mushroom industry.

Shifting partial work from cloud to edge offers SMEs an effective and cost-saving solution to overcome the issues of excessive bandwidth-cloud utilization and high latency [10]. However, determining which features should be conducted in the edge server highly depends on the specific functional requirements and capacity of each smart farming system [19]. This suggests the necessity to design a proper digital Farm Management Information System (FMIS) architecture assisting smart agriculture applications for the specific farmer's requirements (e.g., for each different crop) [20–22], hereafter the Smart Agriculture Information System (SAIS). From the perspective of an SME, Sejong Rain Company in collaboration with Chungnam National University aims to design and implement a low-cost SAIS architecture customized for smart greenhouse mushroom management in association with a paradigm shift to edge computing for business objectives, in this article. This system was tested in a pilot site located in Songsan Green City, Republic of Korea. The advanced Agricultural IoT sensors developed by Korean companies, a standard network flow that is possible for scalability, and edge computing tasks suitable for greenhouse mushroom cultivation, together with examples of how to implement them, were introduced through the proposed SAIS architecture. Finally, a potential SAIS architecture for business purposes in future studies was also suggested, especially for the developing country markets that consider smart agriculture development as the core industry.

2. Background and Related Work

2.1. Smart Agriculture Information System (SAIS)

The traditional FMIS, which relied on simple farm recordkeeping tasks, has been developed since the 1970s to provide useful information for decision-makers to effectively manage farming activities [23]. However, in the era of Agriculture 4.0 nowadays, the adoption of a traditional FMIS is limited when massive data are provided from the IoT sensors and complex tasks are required for modern farm management [24,25]. To cope with the rapid increase of agricultural data pools and data-driven farming applications, the SAIS, whose cornerstone is the large digital FMIS specialized for precision agriculture, can improve the automation and efficiency in managing enormous amounts of farming information including data collection, data processing, and data-driven decision making [20,25]. Although the core of the SAIS is the multi-layered architecture comprising four basic components of Smart Product, Network Connection, Data, and Smart Service [22], recent studies paid more attention to adapting this core SAIS to the domain-specific uses by breaking it into higher-level architectures, ranging from the classical three- or four-layer up to the advanced seven- or eight-layer architectures. Among the classical architectures, the three-layer one, which corresponds to the two-layer without the Network Layer, refers to the Device-Cloud protocol where the raw data obtained by sensors will be directly transmitted to the cloud for storage, processing, and applications. The four-layer architecture (three-layer without the Network Layer) can allow edge computing to involve those cloud tasks. Integrating a separate layer of edge computing into SAIS architectures is also gaining attraction and possibly becoming a trending paradigm in future studies [17]. Additionally, advanced architectures improved on the classical ones for business objectives by proposing enterprise architecture by adding new layers such as the Business, User, or Security Layers [26]. For more details, the readers are referred to the literature reviews of different multi-layered SAISs in previous studies, summarized in Table 1.

Table 1. Literature reviews of multi-layered Smart Agriculture Information Systems.

Architecture	Study	Edge Computing	Layer
Three-layer (Two-layer without Network Layer)	[27]	No	Node, Base Station, Data Center
	[25]	No	Device, Network, Application
	[28]	No	Data Aggregation, Communication, Application
Four-layer (Three-layer without Network Layer)	[29]	Yes	Things, Edge, Communication, Cloud
	[30]	Yes	Sensors and Actuators, Fog Computing Clients and Devices, Platform Controller, Cloud Agent
	[31]	No	Perception, Network, Middleware, Application
Five-layer	[32]	Yes	Crop CPS, Edge Computing, Access Network, Data Cloud, Analytics
Six-layer	[33]	No	Physical, Network, Middleware, Service, Analytics, User Experience
Seven-layer	[20]	No	Device, Network, Session, Application, Business, Management, Security
Eight-layer	[34]	Yes	Data Wrapper, Device Manager, Exploration Module, Data Aggregation, Data Federation, Event Recognition, Real-time Reasoning, Outward Agent

2.2. Smart Greenhouse Mushroom Cultivation

Aside from the benefits for human health and commerce, edible mushrooms are currently becoming the preferred crop for indoor cultivation, even in urban farming, because they are easily relocated, do not require large space and direct sunlight, and need only a short duration until harvesting [35]. However, the major limitation came from the high sensitivity of mushrooms to the surrounding environment, since extreme weather conditions can immediately damage mushroom health. Therefore, the most important task of indoor mushroom cultivation relies on the timely and continuous control of the key environmental parameters' ideal conditions at every mushroom growing stage [36]. Previous studies listed a wide range of those key parameters [13], but the two most important ones that are worth regarding are air temperature and humidity, while several studies further considered light intensity or carbon dioxide (CO_2) levels. The ideal ranges for these parameters vary depending on different mushroom species [7,37], and they can be easily, timely, and automatically observed and controlled in the greenhouse using IoT sensors, which have been widely reported in the literature as the common approach for indoor mushroom management. Alongside the development of computer vision and precision agriculture technologies, high-resolution cameras associated with AI-based image processing have been successfully applied in the mushroom industry to enhance management accuracy and efficiency [38]. Current advances improved by combining the advantages of IoT sensors and camera computer vision, which opens a new door for future smart greenhouse mushroom cultivation in particular and smart agriculture in general. Literature reviews of smart greenhouse mushroom management using IoT sensors, cameras, and integrated IoT sensor-camera systems are summarized in Table 2.

Table 2. Literature reviews of smart greenhouse mushroom management.

System and Method	Study	Mushroom Species	Monitoring Parameters
IoT environmental sensor	[39]	shiitake	temperature, humidity, CO_2
	[40]	oyster	temperature, humidity, light
	[13]	oyster	temperature, humidity, light
	[36]	gourmet *	temperature, humidity, CO_2, light
	[41]	oyster	temperature, humidity
Camera and AI computer vision	[42]	enoki	RGB image
	[43]	white button	RGB image
	[44]	gourmet	RGB image
	[45]	oyster	RGB image
	[8]	gourmet	RGB image
Integrated IoT environmental sensor-camera	[46]	white button	temperature, humidity, RGB image
	[47]	white button	temperature, humidity, CO_2, RGB image
	[28]	gourmet	temperature, humidity, soil moisture, soil temperature, light, RGB image
	[48]	gourmet	temperature, humidity, RGB image
	[49]	oyster	temperature, humidity, light, soil moisture, RGB image

* gourmet mushrooms include all edible mushrooms such as oyster, shiitake, enoki, or white button mushrooms.

3. System Architecture Design

Based on the potential of the SAIS with edge computing and the increased need for mushroom products, this study aims to design and implement a multi-layered SAIS architecture with leveraging edge computing for cost-effective smart greenhouse mushroom cultivation. In particular, an integrated IoT environmental sensor-camera system and the oyster mushroom were selected to test in this study. To highlight the capability of coupling the edge computing model to the classical Device-Cloud protocol, a general three-layer SAIS architecture (without the Network Layer), which adopted the Edge Layer as a bridge for the Device and Cloud (Device-Edge-Cloud protocol), was employed. Moreover, we also suggested the integration of a bidirectional (two-way) communication mechanism in this system to separate the network flow into two major domains for different functional tasks, including the Forward and Backward Domains. In each layer, two modules were included to respond to separated functional tasks in the Forward and Backward Domains, respectively. The proposed SAIS architecture applied to greenhouse mushroom management in this paper is depicted in Figure 1. Specifically, according to Figure 1, the Forward Domain enables data transmission from the Device Layer (Data Collecting module) through the Edge Layer for quality control and aggregation (Data Preprocessing module), and then to the Cloud Layer (Data Storage module). In contrast, the Backward Domain allows an inverse direction that AI-based solutions developed at the Cloud Layer using stored data (Data Analytics/AI Development module) to be deployed at the Edge Layer for real-time applications (Real-time AI Deployment module), then to control the ideal conditions of the environmental parameters if necessary via controlling actuators/sensors at the Device Layer (Device Controlling module). The Forward Domain is responsible for the data management and storage tasks while the Backward Domain is responsible for the solutions development and decision-making based on the obtained data. In this study, we describe the proposed architecture following its network flow and focus especially on the Forward Domain, since it is the mainstream in the SAIS. It is important to note that although the Network/Communication Layer which encompasses the protocols for network connection is also important and was already employed in this paper, we did not focus on modifying this layer, so it was not mentioned here as a major layer. A detailed description of the proposed SAIS architecture and its implementation for smart greenhouse mushroom management at the testbed site is provided below.

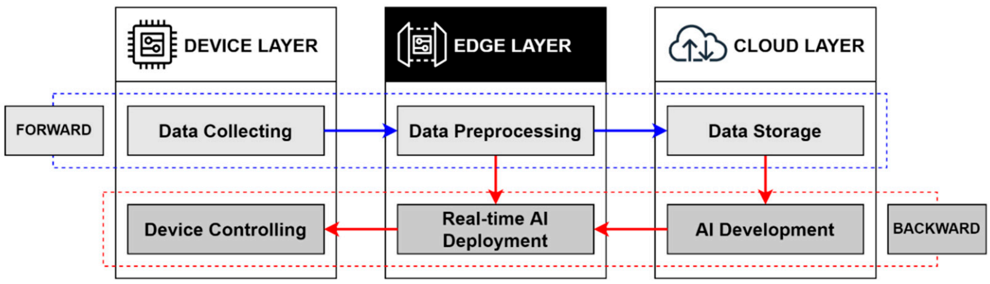

Figure 1. A proposed design of the Smart Agriculture Information System (SAIS) architecture for smart greenhouse mushroom cultivation. The blue dashed border and arrows indicate the Forward Domain and its procedure, and the red dashed border and arrows indicate the Backward Domain and its procedure.

3.1. Forward Domain–Device Layer

The Device Layer is the first layer in this proposed SAIS architecture, which enables a workspace for Agricultural IoT end devices. In this layer, the Data Collecting module is particularly responsible for the Forward Domain. Its detailed architecture is presented in Figure 2.

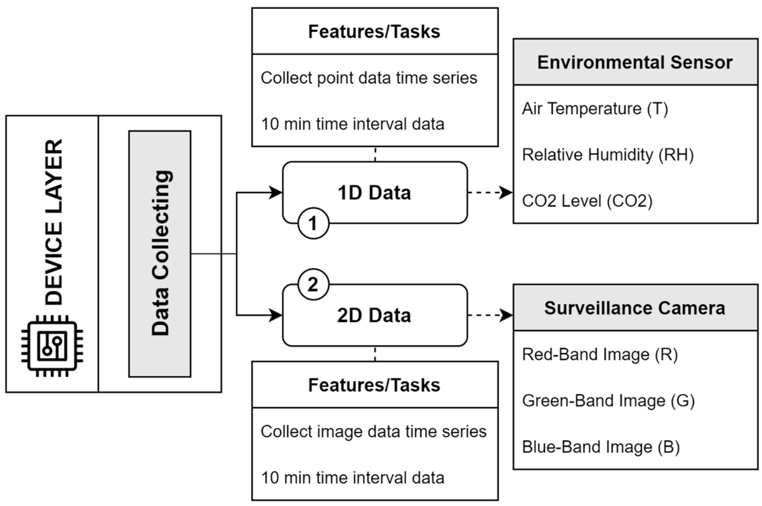

Figure 2. The architecture of the Forward Domain in the Device Layer.

Data Collecting: This module aims to collect farming and environmental data observed from the IoT sensors system. Since the integrated IoT environmental sensor-camera system was employed, this module was separated into two sub-modules for better management, consisting of (1) the one-dimensional (1D) Data sub-module which stands for point environmental data collected from the IoT sensors, and (2) the two-dimensional (2D) Data sub-module, which stands for image data collected from the cameras.

- **1-Dimensional Data (1):** This sub-module offers a workspace for the point IoT sensors to collect the environmental data time series. Because temperature, humidity, and CO_2 are generally the key environmental parameters for mushroom growth in the literature, they were considered in this system. A three-in-one IoT environmental sensor, which measures automatically and simultaneously the three key parameters of temperature (T), relative humidity (RH), and CO_2 levels (CO2) in real-time using only one sensor, was adopted in this study. This low-cost combination sensor is a domestic Korean product that was researched and developed by the Sejong Rain Company, Republic of Korea, and is currently being tested before commercialization. Even though this combination sensor can observe real-time data, the maximum 10 min time resolution has been set to be updated in this sub-module. For more information or inquiries on this combination IoT sensor, please refer to the company homepage (http://sejongrain.com/, accessed on 10 January 2024). Its photos and specifications are shown in Figure 3a,b, and Table 3, respectively.

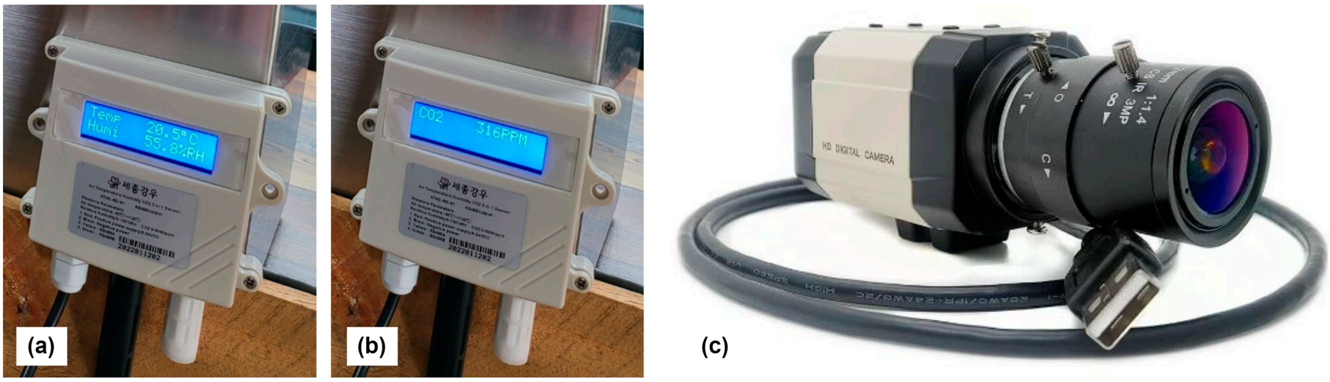

Figure 3. Photos of the combination IoT environmental sensor: (**a**) displaying temperature and humidity and (**b**) displaying CO_2 level; and (**c**) photo of the surveillance camera used in this study. The Korean word in the IoT combination sensor label is the name of the company who develops this device, the Sejong Rain Company, Republic of Korea.

Table 3. Specifications of the combination IoT environmental sensor used.

Specification	Temperature (T)	Humidity (RH)	CO_2 Level (CO2)
Measuring unit	°C	%	ppm
Measuring range	−40–120	0–100	0–2000
Resolution	0.1	0.1	1
Accuracy	±0.2	±3	±20
Stability	maintaining an error of less than 1% throughout the life of the sensor		
Response time	less than 1 s		
Operating condition	−30–70 °C (temperature)/0–100% (humidity)		
Ingress Protection (IP) rating	65 (certificated by the authorized organization)		

- **Two-Dimensional Data (2):** Besides the IoT environmental sensor, this system uses a surveillance camera system to collect crop image/video (time series of RGB images) data, which enhances automation in intuitive recognition of mushroom characteristics (e.g., shapes, colors, species, phenological stages, or related diseases) and was implemented in this sub-module. Specifically, an industrial high-definition (HD 720p) digital camera (model SMT-720PUSBBOX) manufactured by Smtkey Electronic Technology Company, Shenzhen Guangdong, China, was employed to provide streaming images/video of the oyster mushroom. However, to match the temporal resolution of the environmental sensor, the camera system was set to capture image time series at 10 min intervals in single red (R), green (G), and blue (B) bands. A photo of the camera used is shown in Figure 3c and its specifications are presented in Table 4.

Table 4. Specifications of the surveillance camera used.

Surveillance Camera		Specification
Image sensor	Sensor	720P CMOS
	Lens	2.8–12/5–50/6–60 CS Fixed Varifocal Zoom Lens (optional)
	Effective pixel	FHD 1280 (H) × 720 (V)
	Output image format	MJPEG/YUV2 (YUYV)
	Minimum illumination	0.051 lux
Night vision mode		supported (need to cooperate with infrared lens @ 850 or 940 nm)
Operating temperature (°C)		−10–60
Weight (g)		30

3.2. Forward Domain–Edge Layer

The Edge Layer or edge server is a middle layer in this proposed three-layer SAIS, where edge computing is executed, and is the major part of this study. It is important to note that edge computing is not a replacement for traditional cloud computing, but that it is a complement for cloud computing. Edge computing is normally conducted in low-memory and low-power microprocessors and assigned for small workloads with lightweight computation, which fits in with real-time AI applications. In contrast, cloud computing is compatible with smart solutions requiring BD and heavy computation capacity. Therefore, a powerful, low-energy-consumption, and cost-effective Raspberry Pi single-board computer was used in this study for edge computing in both Forward and Backward Domains. In particular, the Raspberry Pi 3 Model B+, which also provides a high-speed processor suitable for HD video processing and dual connections via both WiFi (wireless) and Ethernet port (wired), was employed in the system. For a more detailed description of its specifications, users are referred to the company's website (https://www.raspberrypi.com/, accessed on 10 January 2024). The related tasks are shown in the Data Preprocessing module for the Forward Domain, and its architecture is depicted in Figure 4.

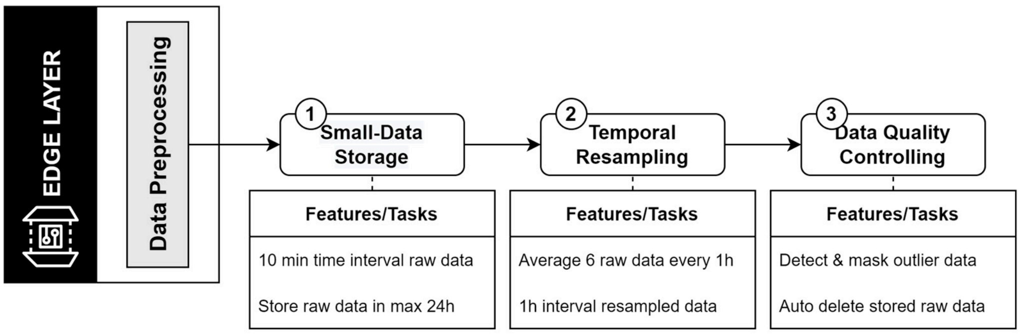

Figure 4. The architecture of the Forward Domain in the Edge Layer.

Data Preprocessing: It is important to note that not all the raw data should be stored in the cloud. Selecting essential data for the cloud can save resources by decreasing cloud and bandwidth workloads. Hence, the main mission of this module is to reduce the number of raw data observed from the Device Layer while keeping data quality before sending them to the Cloud Layer, based on the following sub-modules for preprocessing:

- **Small-Data Storage (1):** Since a previous study indicated the effective greenhouse mushroom monitoring on an hourly basis [13], we also set a one-hour (1 h) interval as the standard temporal resolution to be sent to the cloud and for mushroom cultivation in this study. To this end, several raw data should be stored in the Edge Layer for further temporal data resampling and quality control development.
- **Temporal Resampling (2):** The temporal resampling applied in this sub-module aims to convert the 10 min raw data into hourly data and relies on a simple Average Filtering method, whereas the six raw data samples within every 1h interval stored in the Small-Data Storage sub-module will be transferred to this sub-module for averaging into one data sample. This method can be applied to both 1D and 2D data in this SAIS and may help it reduce the number of data together with overcoming the temporal gaps (missing data) that occurred at the original 10 min interval.
- **Data Quality Control (3):** Despite the benefit of temporal resampling in dealing with the gaps in raw data time series, the resampled data still probably suffer from temporal gaps (when six raw data samples are missing values) or sudden extreme conditions. This requires an AI-based data quality control filter that can not only automatically and continuously detect such outliers in the data stream, but also has a low computational cost when it is applied to the Edge Layer. Even though several lightweight AI models are compatible with edge/fog computing, the well-known k-nearest neighbors (k-NN) algorithm was used for showcasing in this research. The k-NN is simply a non-parametric supervised learning method, which considers k samples of training data to solve the classification and regression problems, but it can be regarded as an unsupervised learning algorithm when it is applied to anomaly detection [50]. The k-NN integrated with a 24 h moving window was applied in this study to identify whether real-time data are anomalous or not, based on the lagged 23h data samples stored in the Small-Data Storage sub-module. Whenever the ksample is detected and masked, it can be continuously used to identify the (k + 1) sample, and the (k − 23) sample is then automatically removed from the Small-Data Storage sub-module. This AI-based data quality control filter can be applied to both 1D and 2D data. For 1D data, besides transmitting them to the Cloud Layer for long-term storage, the processed data were also sent to a responsive module of the Backward Domain in the Edge Layer to support the system's real-time decision-making. However, in the case of 2D data, to reduce the high computational cost when processing image data, the RGB images obtained from the camera were first converted to grayscale images and then transformed to 1D format by simply averaging the digital number (DN) values within an image scene (scene-averaging) before they can be applied, with the data quality control filter. Anomalous DN data samples closer to 0 can be identified as temporal gaps (black images), while those with high values (e.g., higher than 80—a typical average grayscale digital number value) can be classified as light images, which still provide useful information. The quality-controlled image data on an hourly basis were sent only to the Cloud Layer.

3.3. Forward Domain–Cloud Layer

The Cloud Layer is the final layer in this proposed SAIS, which is widely used in various architectures and allows high computational application development based on the quality-controlled BD. Hence, the Forward Domain offers long-term high-quality data to be stored firstly in the Cloud Layer under the Data Storage module management.

Data Storage: Because the accuracy of AI models significantly depends on the quality and quantity of the training data, this module aims to provide a space to store long-term (e.g., one life cycle of mushrooms) high-quality data observed from the integrated environmental sensor-camera system to develop AI-based solutions which require a high computation cost. These necessary datasets stored in this module can be then transmitted to the responsive module of the Backward Domain in the Cloud Layer for training AI models with respect to specific user requirements.

3.4. Backward Domain–Cloud Layer

As highlighted in the description above, the mission of the Backward Domain is to develop AI-based smart solutions and make decisions based on the collected and stored data in the Forward Domain. Therefore, in the Cloud Layer, the Backward Domain tasks are conducted in the AI Development module. Figure 5 presents the architecture of the Backward Domain in the Cloud Layer.

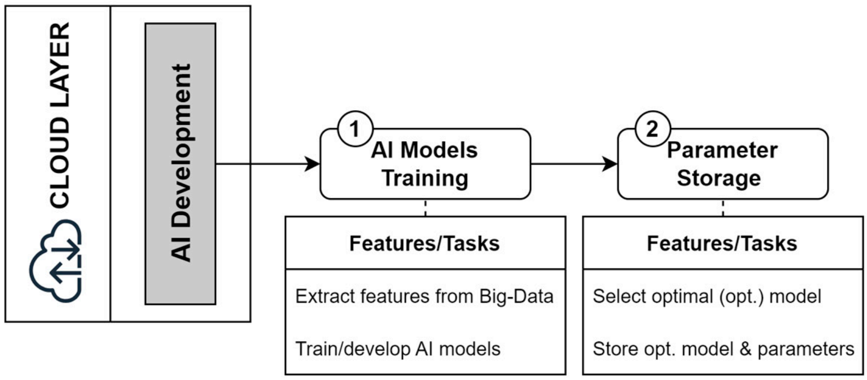

Figure 5. The architecture of the Backward Domain in the Cloud Layer.

AI Development: This module is designed to account for AI-based smart solution development for specific demands, which requires BD for high computational cost training and complex analytics. The implementation of this module in the Backward Domain of the Cloud Layer follows the two sub-modules:

- **AI Model Training (1):** This sub-module includes the feature selection task for training AI models based on user-specific demands, whereas relevant important features were extracted from the BD in the Data Storage module. Furthermore, several well-known AI model candidates will be selected for training.
- **Parameter Storage (2):** The optimal AI model among the candidates was chosen and its optimal parameters were then stored in this sub-module for further deployment.

Various AI-based smart greenhouse mushroom-cultivation solutions can be considered for development. However, to showcase only one example of the high-computational-cost AI model application with BD (mostly with image/video data) in the Cloud Layer, a well-known, fast, effective, and advanced computer vision algorithm version among the series of the "You Only Look Once" (YOLO) framework [51], the YOLOv5 model, was used in this study for real-time mushroom detection with a surveillance camera. Specifically, the YOLOv5 model, whose cornerstone is a simple deep convolutional neural network, was introduced in 2020 to improve on its predecessors in object detection. The YOLOv5 model divides images into a grid cell and finally predicts the objects with multiple bounding boxes per grid cell. Its architecture consists of three main components: the BackBone for feature extraction, the Neck for feature fusion, and the Head for feature conversion to bounding-box parameters. The BD images stored in the Big-Data Storage module can be labeled for training the YOLOv5 model. To reduce the training costs (including the training time and number of training-data samples), this study adopted the Transfer Learning method to fine-tune the pre-trained YOLOv5 model for mushroom detection to be compatible with the current training mushroom images in the Big-Data Storage module. The optimal model parameters were stored in the Parameter Storage sub-module and then transferred to the Edge Layer for real-time deployment.

3.5. Backward Domain–Edge Layer

The Backward Domain in the Edge Layer provides the workspace for real-time smart agriculture solution deployment, which was assigned to the Real-time AI Deployment module, with its architecture provided in Figure 6.

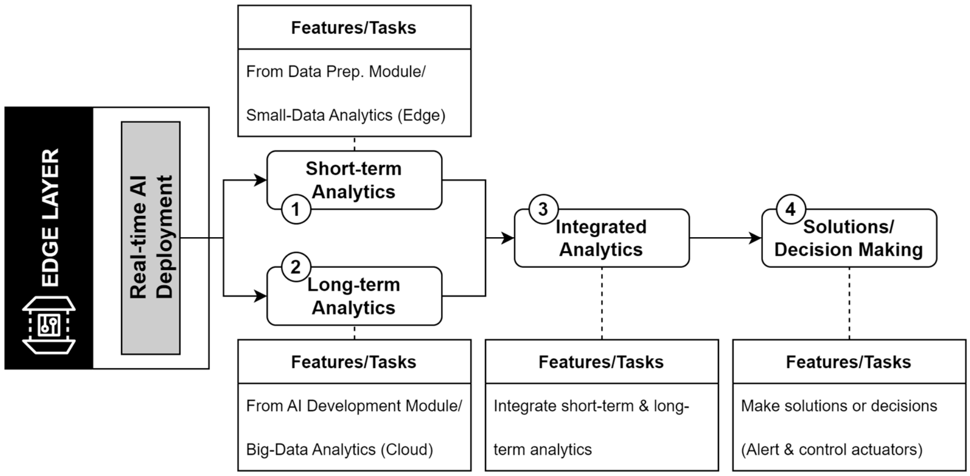

Figure 6. The architecture of the Backward Domain in the Edge Layer.

Real-time AI Deployment: This module consists of four sub-modules for integrating analytics from both environmental sensors and cameras to improve real-time solution/decision making, as follows:

- **Short-term Analytics (1):** This sub-module receives the analytical results from the Forward Domain in the Edge Layer (Data Preprocessing module), mainly based on temporary small-data storage of the 1D data from the IoT environmental sensor.
- **Long-term Analytics (2):** The optimal AI model obtained from the Backward Domain based on BD analytics in the Cloud Layer (AI Development module) was deployed in this sub-module, mainly for the 2D image data.
- **Integrated Analytics (3):** This sub-module combines the analytical results from the Short-term and Long-term Analytics sub-modules to enhance decision-making.
- **Solutions/Decision Making (4):** Relevant solutions/decisions will be made in this sub-module, based on the results from the Integrated Analytics sub-module.

3.6. Backward Domain–Device Layer

The Backward Domain in the Device Layer covers the Device Controlling module.

Device Controlling: This module aims to automatically control sensors/actuators to make farm management decisions as soon as possible, based on the analytical solutions provided by the Real-time AI Deployment module in the Edge Layer. For example, the image-based mushroom detection solution allows users to recognize whether the mushrooms are currently in the growing period or have been harvested. In the case of extreme conditions observed by the environmental sensor during the mushroom growing period, this module enables the modulating of key environmental parameters by controlling IoT sensors/actuators such as the heater/cooling fan (for temperature), humidifier/dehumidifier (for humidity), or air ventilation (for CO_2 level) via connection to a microcontroller.

4. Implementation Results

4.1. Implementation of Forward Domain–Device Layer

Installation of the integrated IoT environmental sensor-surveillance camera system for smart greenhouse mushroom cultivation is the first implementation step and was carried out in a mushroom house located in Songsan Green City, Republic of Korea. This system aims to collect the key environmental parameters affecting mushroom growth and intuitive mushroom information, which is the responsibility of the Data Collecting module belonging to the Forward Domain in the Device Layer. Figure 7 describes the installation of the combination IoT environmental sensor for simultaneously measuring the real-time 1D data of three key environmental parameters including air temperature (T), atmospheric relative humidity (RH), and CO_2 level (CO2), together with the construction of a data logger containing the Raspberry Pi single-board computer. Data collected from the IoT environmental sensor were then transmitted to the data logger for further processing, mainly via a wireless connection (Figure 7a). However, we also set up ethernet cables here to ensure that data transmission can be automatically switched to a wired connection when the wireless connection collapses, maintaining their stable transmission (Figure 7b). It is important to note that although the environmental sensor was set up in the greenhouse environment, we constructed the data logger outside the mushroom house for future combination with outdoor smart agriculture sensors/actuators such as smart precipitation gauges, unmanned aerial vehicles, tractors, and so on (Figure 7c).

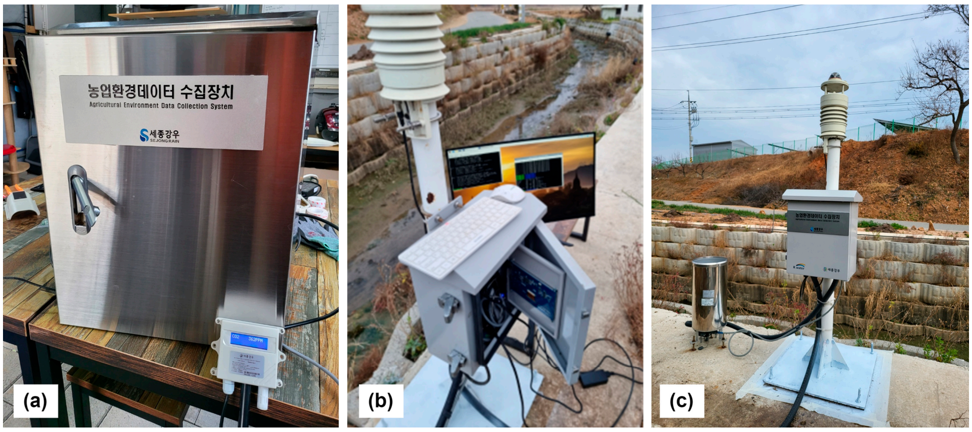

Figure 7. Installation of the combination IoT environmental sensor for the greenhouse mushroom information system with connection to data logger via the following: (**a**) wireless connection and (**b**) wired connection. (**c**) Construction of data logger outdoors with potential combination with outdoor sensors. The Korean words in the label of the data logger represent for the system name "Agriculture Environment Data Collection System" and the company name "Sejong Rain".

The time series of the raw 1D data (at 10 min of temporal resolution) for the three key parameters from the installed IoT environmental sensor for nearly one month (one life cycle of the mushroom) is depicted in Figure 8. In general, no major operational errors occurred when continuous data were collected without missing values and the observed data range fell within the sensor's allowable measuring range for each parameter, according to Table 3. This suggests that the combination IoT environmental sensor used in this study, which was developed by the Korean company, is ready and has potential for smart mushroom monitoring and further applications in smart agriculture.

Besides the IoT environmental sensor, the simultaneous installation of the surveillance camera to capture real-time intuitive 2D data for crop information was also implemented in the pilot area, as illustrated in Figure 9. The camera was fixed to collect a time series of RGB images covering a partial mushroom farm for preliminary testing. We also connected the camera system to the data logger for data transmission.

Time series of the raw 2D data at 10 min time intervals were obtained from the installed surveillance camera. We showcase several RGB image samples of mushrooms at different growth stages in Figure 10. It can be drawn from Figure 10 that intuitive mushroom information can be well captured by the installed surveillance camera, even with the changes between growth stages. However, there are several temporal gaps (image samples with no-value/black images) that occurred during the operation which need to be noticed and addressed. All the raw data collected in the Device Layer were then sent to the Edge Layer for pre-processing.

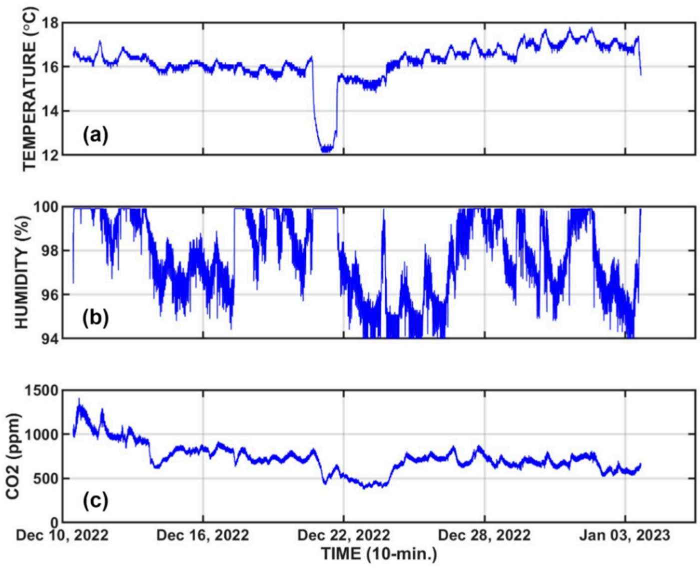

Figure 8. Time series of 1D raw data in 10 min intervals collected from the combination IoT environmental sensor for (**a**) temperature (T), (**b**) humidity (RH), and (**c**) CO_2 level (CO2).

Figure 9. Installation of the surveillance camera for the greenhouse-mushroom information system.

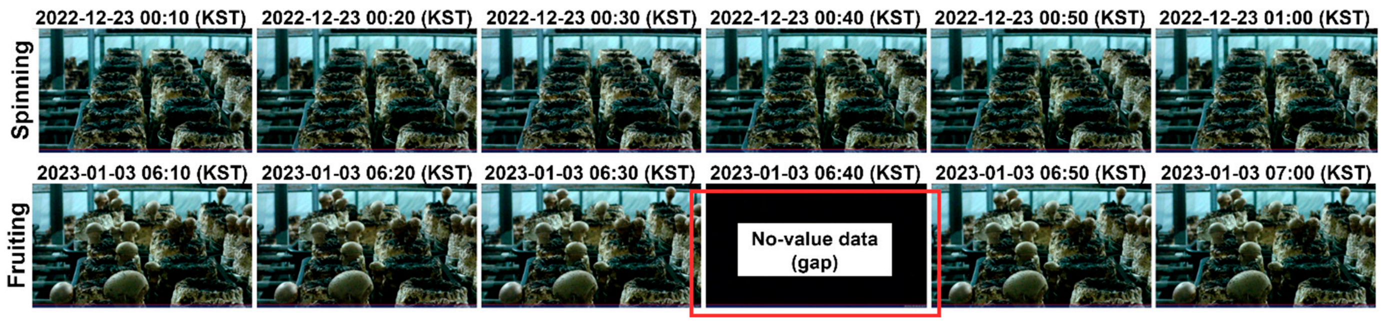

Figure 10. Samples of 2D raw data in 10 min intervals collected from the surveillance camera during different mushroom growth stages. The red rectangle highlights the gap (no-value data) in the data stream.

4.2. Implementation of Forward Domain–Edge Layer

The raw 1D and 2D data collected at the Device Layer were then delivered to the Forward Domain in the Edge Layer for further preprocessing before being transmitted to the Cloud Layer. The Edge Layer is the major layer in this designed SAIS, whose main goal is a stepping stone toward saving cloud and bandwidth resources as well as selecting essential data for the cloud server. Therefore, the Forward Domain in this layer (represented by the Data Preprocessing module) is responsible for reducing unnecessary data collected from the Device Layer and conducting quality control for high-quality data sent to the Cloud Layer. To this end, several raw data samples (small data) should be stored in this layer for a short period and the Small-Data Storage sub-module is responsible for providing storage space in this task.

The first preprocessing task in this module is data reduction. Since the mushroom monitoring on an hourly basis is sufficient and was selected as the standard temporal resolution to be updated in the cloud, an average filtering technique was utilized as the temporal resampling method to convert 10 min data into 1h data. Figure 11 illustrates examples of how to conduct this temporal resampling method in both 1D and 2D data time series. In particular, every six raw data samples/images within a current 1h duration were initially stored. Since no significant differences were observed among these raw data, an hourly representative point value/image can be generated by simply taking an average of those six data samples/images. After an hourly representative data sample/image is given, it will be sent back to the Small-Data Storage sub-module for temporal storage and the current raw data will be automatically removed from this sub-module for updating the coming raw data. It is important to note that, besides the data reduction benefit, this method also supports dealing with temporal gap problems, ensuring continuous measurement of the data stream. For example, a black image (temporal gap) that occurred in the raw 2D data was successfully removed and replaced by a representative image (average image of the six raw data in 10 min intervals within every 1h interval) while keeping sufficient information, as illustrated in Figure 11b.

The second preprocessing task in the Forward Domain of the Edge Layer is for data quality control, which is covered by the Data Quality Controlling sub-module. For the 1D data, since the important task of smart greenhouse mushroom cultivation is mainly based on modulating the ideal conditions of key environmental parameters, this sub-module aims to determine the sudden extreme conditions occurring in the environmental data, supporting the system with timely responses to address the problems as well as managing high-quality data sent to the cloud server. However, because edge computing is compatible with light computing applications that can work well in short-term small-data numbers stored in this layer, effective and lightweight AI algorithms are preferable for this task. As a result, a widely used lightweight AI model, the k-NN algorithm, was selected as the anomaly-detection method in this study. More specifically, the k-NN was conducted in a 24h moving window (k = 23) for the continuous hourly data samples stored in the Small-Data Storage. The real-time outlier detection of the current data sample (the k sample) can be performed based on its recent 23 lagged hourly data samples. Whenever the k sample is detected and masked, it can be continuously used to identify the (k + 1) sample, and the (k − 23) sample is automatically deleted from the Small-Data Storage sub-module. Figure 12 presents the results of applying the k-NN algorithm-based anomaly detection to the 1D hourly data of three key environmental parameters for mushroom cultivation. These successful implementation results of the k-NN model-based anomaly detection in the 1D environmental data, as illustrated in Figure 12, highlight the suitability and potential application of lightweight AI models for edge computing.

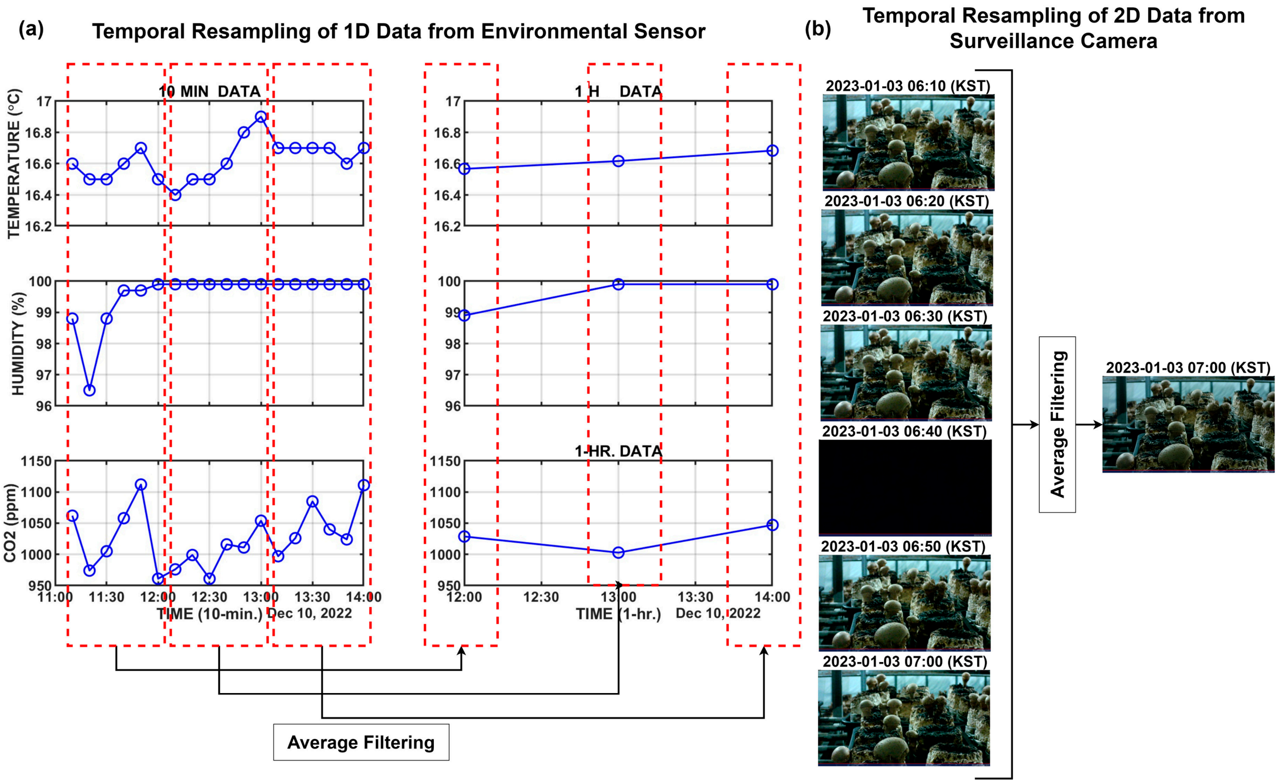

Figure 11. Examples of the temporal resampling based on the average filtering method to resample raw data in 10 min intervals into standard data in 1h intervals applied to the following: (**a**) 1D data from the environmental sensor and (**b**) 2D data from the camera. The red dashed rectangle represents the 1h data window.

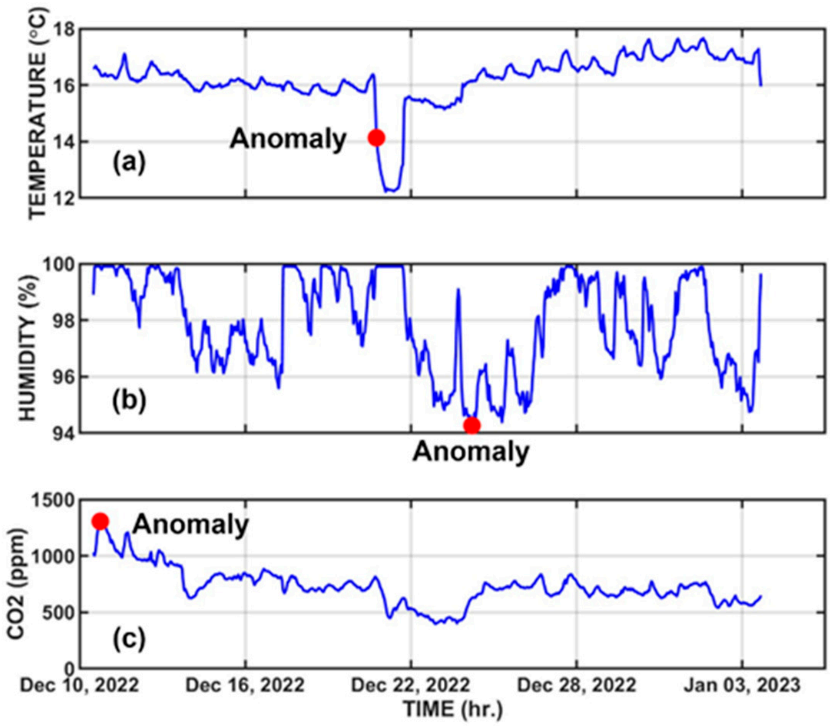

Figure 12. Results of the AI-based anomaly-detection application to the 1D hourly (hr) environmental data for (**a**) temperature (T), (**b**) humidity (RH), and (**c**) CO_2 level (CO2).

As regards the 2D data, one problem is that although the temporal resampling method can deal with temporal gaps in the raw data, they still possibly occur in the resampled data (when the six original images within one hour are all black images). Hence, the anomaly-detection method should be applied to the hourly 2D data to detect these gaps. However, to deal with the high computational cost when processing numerous crop images, a preprocessing procedure series was applied to the 2D data to transform them into 1D data before conducting the AI-based anomaly detection. In particular, an RGB image was first converted to a grayscale image and subsequently transformed into a digital number value by averaging grayscale digital numbers from all pixels within an image scene (scene averaging). Figure 13 provides a general example of how to carry out AI-based anomaly detection in the 2D data and their results for mushroom images in this study. As can be seen from Figure 13, two types of outliers have been identified by using the k-NN anomaly-detection method. Anomalous data with digital numbers closer to 0 are assigned to the temporal gaps (black images), which do not provide any useful information, whereas the outliers with the highest values (e.g., greater than 80—a typical average grayscale digital number value) are identified and masked as "light image", which still provide similar crop information to the normal images but at a higher brightness. Finally, after the resampled data are quality-controlled, they will be sent to the cloud server for storage and to the Short-term Analytics sub-module in the Backward Domain of the Edge Layer for supporting real-time decision making.

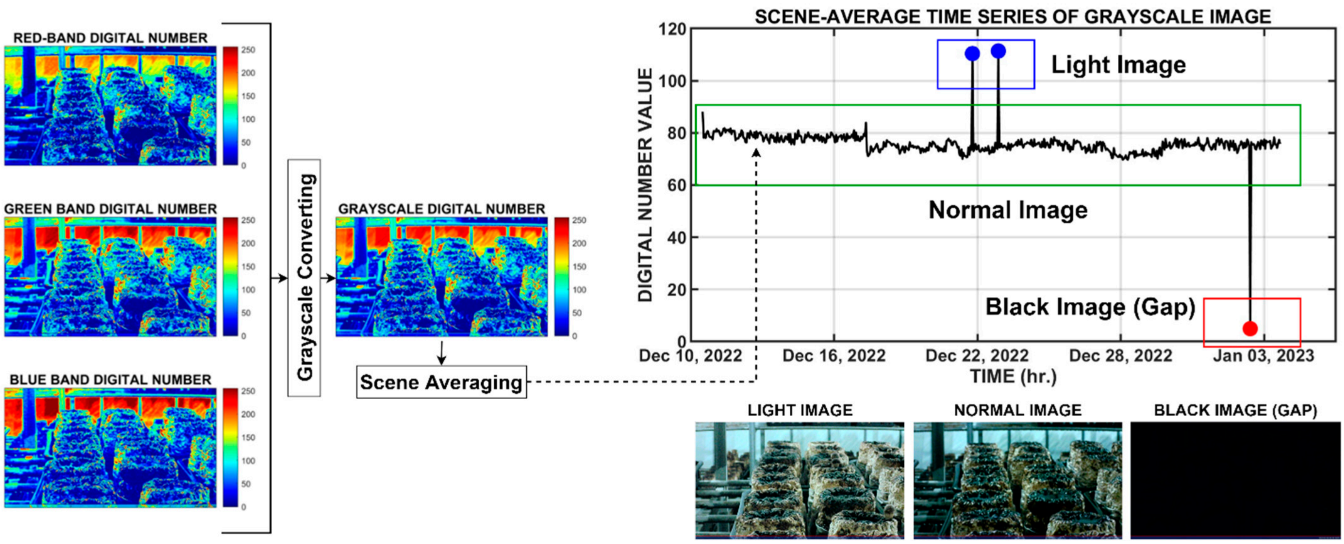

Figure 13. An example of the AI-based anomaly-detection application to the 2D image data. The color maps on the left side display the heatmap of an original image in digital numbers for red, green, blue, and grayscale channels.

4.3. Implementation of Forward- and Backward-Domain–Cloud Layer

In the Cloud Layer, high-quality BD transmitted from the Edge Layer after passing the data quality control procedure will be initially stored in the Big-Data Storage Module of the Forward Domain. These high-quality BD stored in the Cloud Layer were then sent to the AI Development Module for training high-computational-cost AI models based on specific user requirements. This study employed a widely used deep learning-based object-detection algorithm, the YOLOv5 model, to address the mushroom recognition problem, mainly for the 2D image data. Moreover, to reduce the computational cost during the training process, the transfer learning technique was integrated to fine-tune the pre-trained YOLOv5 model for adapting the mushroom images stored in the Cloud Layer. Detailed information on training and evaluation of the YOLOv5 model for mushroom detection in this study is shown in Table 5. The obtained optimal hyperparameters were subsequently stored in the Parameter Storage module and were used for real-time deployment of this solution in the Edge Layer.

Table 5. Information on the training and evaluation of the YOLOv5 model for mushroom detection.

Training		Evaluation	
No. of samples	402	No. of samples	118
Optimizer	Stochastic Gradient Descent	Precision	0.97
Loss function	Cross entropy	Recall	0.98
Learning rate	0.01	Average Precision (AP)	0.99
Batch size	16	Mean Average Precision (mAP)	0.77
Epoch	100		

4.4. Implementation of Backward Domain–Edge Layer

The Backward Domain in the Edge Layer, which was carried out by the Real-time AI Development module, aims to provide integrated analytics for decision-making by combining the results of short-term and long-term analytics. In particular, on the one hand, the Short-term Analytics module in this layer obtained the key environmental-parameter anomaly-detection results from the Forward Domain in the Edge Layer, which was described in the section above (Implementation of Forward Domain–Edge Layer) and is shown in Figure 12. On the other hand, when it comes to long-term analytics, this module adapted the developed YOLOv5-based mushroom-recognition model for detection, based on the stored optimal model hyperparameters from the Backward Domain in the Cloud Layer. Representative examples of the YOLOv5 model deployment for mushroom detection in this study are depicted in Figure 14, whereas single-object (mushroom) areas are detected in continuous images with the red bounding box and confidence level by further applying instance segmentation. According to the figure, in general, most of the mushroom areas during the fruiting period can be well identified by the system, with a clear discrimination of the non-mushroom areas observed. This indicated the successful deployment of a high-computational-cost AI model in the Edge Layer, demonstrating its suitability to adopt a developed AI model in the Cloud Layer for edge computing. However, there are still several mushroom areas in an image that the system cannot detect, suggesting the need for retraining the model to improve its performance by the updated high-quality BD sent to the Cloud. The results from the long-term analytics can be integrated with those from the short-term analytics to support the system's decision-making. For example, if outliers are detected in the streaming environmental data (results from the short-term analytics) during the mushroom's fruiting period (results from the long-term analytics), the system can automatically send an alert notification to the users or control the sensors/actuators for maintaining the normal status of environmental conditions.

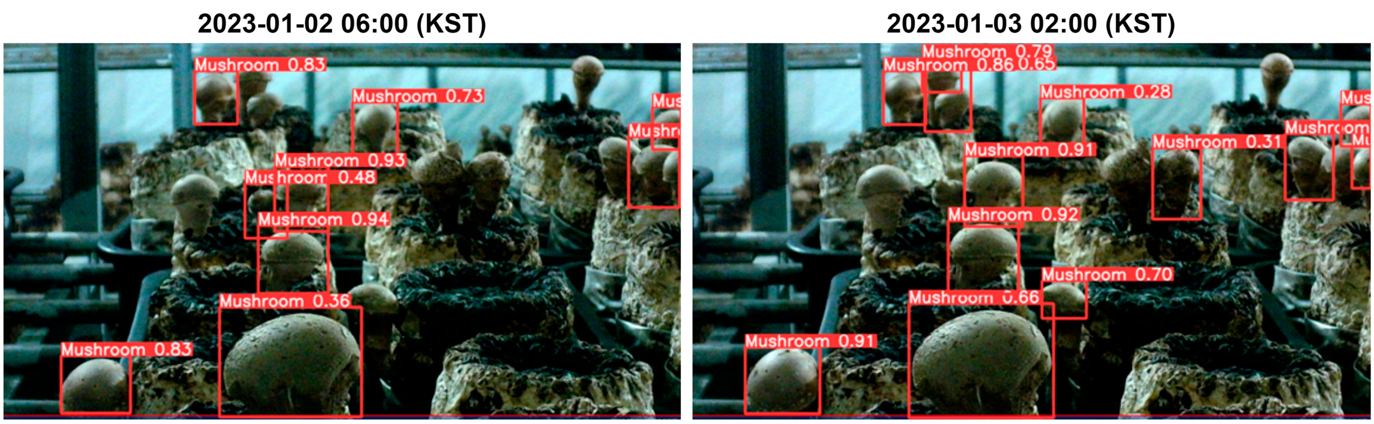

Figure 14. An example of implementation of the Real-time AI Deployment module (deployment of YOLOv5 and transfer learning in mushroom detection).

5. Discussion and Future Direction

The major difference between this proposed research and related work mentioned in Tables 1 and 2 is the combined use of the edge-computing paradigm in a smart greenhouse mushroom-cultivation information system. In particular, the Edge Layer was not included in the SAIS architecture for mushroom cultivation in Table 1 [28], while none of the previous mushroom management studies in Table 2 employed the edge computing model in their system. Thus, the successful practical application of this study can shed new light on leveraging edge-computing paradigm shifts as well as improving smart mushroom cultivation systems. In addition, a minor discrepancy that is worth mentioning is the introduction of a new low-cost combination IoT environmental sensor made by a Korean company. This product combines three single sensors for different environmental parameters into one sensor (three-in-one sensor) and also employs low-cost Raspberry Pi single-board computers working in the resource-saving edge computing paradigm, so it can contribute to the diverse selections for clients in low-cost smart agriculture devices and information system markets.

Notwithstanding the benefits of edge computing in complementing classical cloud computing and in improving smart greenhouse mushroom cultivation, this preliminary study may suffer from several challenges in the future. First, the implementation costs when scaling this framework up to a larger scale for other crops, together with integrating outdoor smart agriculture systems, are questionable, since additional costs from new IoT devices and their installation might be included in the system. This may limit the commercialization of this framework for business objectives. Second, the improvement in mushroom production efficiency when the edge computing paradigm is included is not fully analyzed, suggesting the implementation of more detailed analyses for this problem in further studies.

Addressing potential challenges is also related to the future outlook of this research. Besides the application customized for mushroom cultivation, the successful practical implementation of this study suggests a potential to scale up this framework for business objectives and other user-specific requirements (e.g., other crops) in future studies. To this end, in the future direction, this fundamental SAIS architecture is expected to be coupled with the two new layers, including the User and Administration Layers. The potential architecture for future the SAIS is illustrated in Figure 15, and a detailed description for each added layer is provided below.

- **User Layer:** This layer is particularly designed for the users of the SAIS such as farmers or clients. From the user's perspective, the acquisition of quality-controlled datasets and the intuitive visualization of these datasets with timely notifications if any anomalous events occur from the Cloud Layer, and the capability to control the devices manually in an emergency in the Device Layer, are preferable. Thus, several respective modules such as Data Extraction, Visualization/Notification, or Manual Device Controlling Modules can be considered in this layer. Since the users mainly work in the Device and Cloud Layers, the users can only access these two layers, and they do not need to access the Edge Layer to adjust any edge computing tasks.
- **Administration Layer:** This layer is particularly designed for the providers or developers (e.g., companies) of the SAIS for business goals. Unlike the users, the administrators aim to manage all the data flow from the Device Layer to the Cloud Layer, so they can access and collect data from all the layers in this system, which can be conducted by the Data Extraction/Management module. Moreover, the administrators can meet the specific user requirements by developing suitable solutions, and ensure the safety of the system by maintaining the security. Therefore, the two respective modules including Application Development and Network Security are considered in this layer.

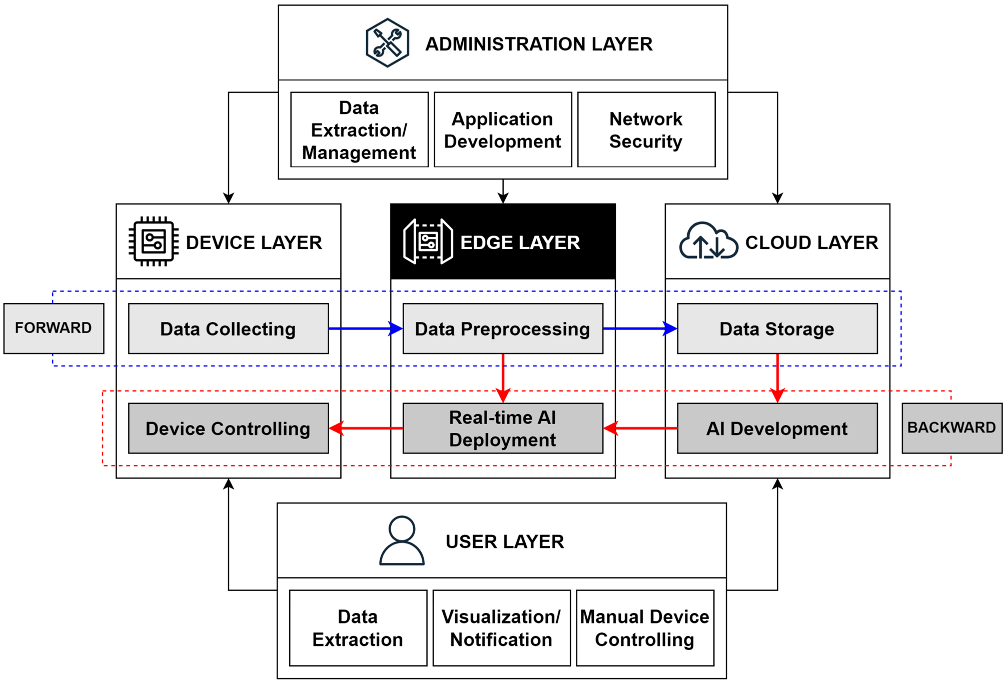

Figure 15. The potential architecture of the future SAIS for business objectives. The blue dashed border and arrows indicate the Forward Domain and its procedure, and the red dashed border and arrows indicate the Backward Domain and its procedure.

Not only are the above additional layers being included, but the Device Layer is also planning to be extended with new sensors, in the future system. Microcontrollers and several controlling sensors/actuators (e.g., heaters, fans, humidifiers, or air ventilation. . .) are expected to be utilized, which can complement the current indoor system. Furthermore, a parallel SAIS for outdoor smart agriculture with relevant IoT devices (e.g., smart rain gauges, automated weather sensors, irrigation valves, or unmanned aerial vehicles. . .) can be considered and developed for an integrated indoor–outdoor smart agriculture system.

6. Conclusions

Traditional Device-Cloud protocol in smart agriculture often suffers from the challenges of delayed system responses caused by low bandwidth and high latency, together with high cloud-service costs for data computation and storage when enormous numbers of data acquired from IoT devices are directly transmitted to the cloud server without processing. Novel edge computing offers an effective solution to deal with challenges in this traditional protocol by shifting partial data storage and computation capability from the cloud server to edge devices. Nevertheless, selecting which tasks can be implemented in edge computing depends on user-specific demands, suggesting the need to design a specific and proper Smart Agriculture Information System (SAIS) architecture that is compatible with single-crop requirements.

Based on the nutritious and commercial benefits of edible mushrooms as well as the necessity of cost reduction from a business viewpoint, the major goal of this study is to design and implement a standard multilayered SAIS architecture for smart greenhouse mushroom cultivation toward leveraging edge computing, which can be scalable for business goals. In this designed SAIS, the three-layer architecture, which couples edge computing in the central layer to connect the Device and Cloud Layers (Device-Edge-Cloud protocol) and enables automation in mushroom management using an integration of the IoT environmental sensor (for mushroom key environmental-parameter monitoring) and surveillance camera (for intuitive mushroom monitoring) was employed and tested in a testbed site in the Republic of Korea. Via this designed SAIS, we aim to introduce the advanced combination environmental IoT sensors developed by Korean companies, a standard network flow that is possible for scalability and suitable edge computing tasks, with typical examples. Moreover, a potential SAIS architecture with additional layers for a future direction for business purposes was also suggested.

The successfully implemented SAIS architecture indicated that our combination IoT environmental sensor integrated with a surveillance camera could monitor the real-time key environmental parameters affecting mushroom growth and intuitive mushroom information. Typical examples of mushroom cultivation based on the collected data revealed that several low-cost data pre-processing procedures including small-data storage, temporal resampling-based data reduction, and lightweight artificial intelligence (AI)-based data quality control for anomaly detection within environmental conditions, together with real-time AI model (YOLOv5) deployment for mushroom recognition from crop images, are compatible with edge computing. In contrast, high-quality BD storage and high-computational-cost AI model development can be implemented in the cloud. Moreover, integrating the Edge Layer as the center of the traditional protocol can significantly save network resources and operational costs by reducing unnecessary data sent from the device to the cloud while keeping sufficient information. Finally, the future improvement suggests including additional layers to meet user-specific demands for business objectives and the extension of the current system with new controlling sensors and a parallel outdoor system, toward an integrated indoor–outdoor smart agriculture system.

Author Contributions: Conceptualization, H.H.N., D.-Y.S. and D.-H.L.; methodology, H.H.N. and D.-Y.S.; software, H.H.N.; validation, D.-Y.S., W.-S.J. and T.-Y.K.; formal analysis, T.-Y.K.; investigation, W.-S.J.; resources, D.-Y.S.; data curation, H.H.N.; writing—original draft preparation, H.H.N.; writing—review and editing, D.-Y.S., W.-S.J. and T.-Y.K.; visualization, H.H.N.; supervision, D.-H.L.; project administration, D.-H.L.; funding acquisition, D.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the "Regional Innovation Strategy" (RIS) through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (MOE) (2021RIS-004), and the research fund of Chungnam National University (CNU), Republic of Korea (2022-0694-01). The APC was also funded by the MOE through the NRF RIS project (2021RIS-004) and the CNU research fund (2022-0694-01).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Authors Hoang Hai Nguyen and Dae-Yun Shin were employed by the company Sejong Rain Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Ba, D.M.; Ssentongo, P.; Beelman, R.B.; Muscat, J.; Gao, X.; Richie, J.P. Higher Mushroom Consumption Is Associated with Lower Risk of Cancer: A Systematic Review and Meta-Analysis of Observational Studies. *Adv. Nutr.* **2021**, *12*, 1691–1704. [CrossRef] [PubMed]
2. Roncero-Ramos, I.; Delgado-Andrade, C. The Beneficial Role of Edible Mushrooms in Human Health. *Curr. Opin. Food Sci.* **2017**, *14*, 122–128. [CrossRef]
3. Hamza, A.; Ghanekar, S.; Santhosh Kumar, D. Current Trends in Health-Promoting Potential and Biomaterial Applications of Edible Mushrooms for Human Wellness. *Food Biosci.* **2023**, *51*, 102290. [CrossRef]
4. Kour, H.; Kour, D.; Kour, S.; Singh, S.; Jawad Hashmi, S.A.; Yadav, A.N.; Kumar, K.; Sharma, Y.P.; Ahluwalia, A.S. Bioactive Compounds from Mushrooms: Emerging Bioresources of Food and Nutraceuticals. *Food Biosci.* **2022**, *50*, 102124. [CrossRef]
5. Kauserud, H.; Stige, L.C.; Vik, J.O.; Økland, R.H.; Høiland, K.; Stenseth, N.C. Mushroom Fruiting and Climate Change. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 3811–3814. [CrossRef]
6. Procházka, P.; Soukupová, J.; Tomšík, K.; Mullen, K.J.; Čábelková, I. Climatic Factors Affecting Wild Mushroom Foraging in Central Europe. *Forests* **2023**, *14*, 382. [CrossRef]
7. Rukhiran, M.; Sutanthavibul, C.; Boonsong, S.; Netinant, P. IoT-Based Mushroom Cultivation System with Solar Renewable Energy Integration: Assessing the Sustainable Impact of the Yield and Quality. *Sustainability* **2023**, *15*, 13968. [CrossRef]

8. Sujatanagarjuna, A.; Kia, S.; Briechle, D.F.; Leiding, B. MushR: A Smart, Automated, and Scalable Indoor Harvesting System for Gourmet Mushrooms. *Agriculture* **2023**, *13*, 1533. [CrossRef]
9. Elijah, O.; Rahman, T.A.; Orikumhi, I.; Leow, C.Y.; Hindia, M.N. An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges. *IEEE Internet Things J.* **2018**, *5*, 3758–3773. [CrossRef]
10. Pan, J.; McElhannon, J. Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet Things J.* **2018**, *5*, 439–449. [CrossRef]
11. Alharbi, H.A.; Aldossary, M. Energy-Efficient Edge-Fog-Cloud Architecture for IoT-Based Smart Agriculture Environment. *IEEE Access* **2021**, *9*, 110480–110492. [CrossRef]
12. Singh, R.; Gill, S.S. Edge AI: A Survey. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 71–92. [CrossRef]
13. Islam, M.A.; Islam, M.A.; Miah, M.S.U.; Bhowmik, A. An Automated Monitoring and Environmental Control System for Laboratory-Scale Cultivation of Oyster Mushrooms Using the Internet of Agricultural Thing (IoAT). In Proceedings of the 2nd International Conference on Computing Advancements, Dhaka, Bangladesh, 10–12 March 2022; pp. 207–212. [CrossRef]
14. Das, R.; Inuwa, M.M. A Review on Fog Computing: Issues, Characteristics, Challenges, and Potential Applications. *Telemat. Inform. Rep.* **2023**, *10*, 100049. [CrossRef]
15. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
16. Zhang, X.; Cao, Z.; Dong, W. Overview of Edge Computing in the Agricultural Internet of Things: Key Technologies, Applications, Challenges. *IEEE Access* **2020**, *8*, 141748–141761. [CrossRef]
17. Lin, Y.B.; Chen, W.E.; Chang, T.C.Y. Moving from Cloud to Fog/Edge: The Smart Agriculture Experience. *IEEE Commun. Mag.* **2023**, *61*, 86–92. [CrossRef]
18. da Costa Bezerra, S.F.; Filho, A.S.M.; Delicato, F.C.; da Rocha, A.R. Article Processing Complex Events in Fog-Based Internet of Things Systems for Smart Agriculture. *Sensors* **2021**, *21*, 7226. [CrossRef]
19. Mansouri, Y.; Babar, M.A. A Review of Edge Computing: Features and Resource Virtualization. *J. Parallel Distrib. Comput.* **2021**, *150*, 155–183. [CrossRef]
20. Köksal; Tekinerdogan, B. Architecture Design Approach for IoT-Based Farm Management Information Systems. *Precis. Agric.* **2019**, *20*, 926–958. [CrossRef]
21. Murakami, E.; Saraiva, A.M.; Ribeiro, L.C.M.; Cugnasca, C.E.; Hirakawa, A.R.; Correa, P.L.P. An Infrastructure for the Development of Distributed Service-Oriented Information Systems for Precision Agriculture. *Comput. Electron. Agric.* **2007**, *58*, 37–48. [CrossRef]
22. Strobel, G. Farming in the Era of Internet of Things: An Information System Architecture for Smart Farming. *WI2020 Community Tracks* **2020**, 208–223. [CrossRef]
23. Fountas, S.; Carli, G.; Sørensen, C.G.; Tsiropoulos, Z.; Cavalaris, C.; Vatsanidou, A.; Liakos, B.; Canavari, M.; Wiebensohn, J.; Tisserye, B. Farm Management Information Systems: Current Situation and Future Perspectives. *Comput. Electron. Agric.* **2015**, *115*, 40–50. [CrossRef]
24. Sørensen, C.G.; Fountas, S.; Nash, E.; Pesonen, L.; Bochtis, D.; Pedersen, S.M.; Basso, B.; Blackmore, S.B. Conceptual Model of a Future Farm Management Information System. *Comput. Electron. Agric.* **2010**, *72*, 37–47. [CrossRef]
25. Villa-Henriksen, A.; Edwards, G.T.C.; Pesonen, L.A.; Green, O.; Sørensen, C.A.G. Internet of Things in Arable Farming: Implementation, Applications, Challenges and Potential. *Biosyst. Eng.* **2020**, *191*, 60–84. [CrossRef]
26. Winter, R.; Fischer, R. Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. In Proceedings of the 2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06), Hong Kong, China, 16–20 October 2006; pp. 30–38. [CrossRef]
27. Li, X.H.; Cheng, X.; Yan, K.; Gong, P. A Monitoring System for Vegetable Greenhouses Based on a Wireless Sensor Network. *Sensors* **2010**, *10*, 8963–8980. [CrossRef] [PubMed]
28. Moysiadis, V.; Karaiskou, C.; Kokkonis, G.; Moscholios, I.D.; Sarigiannidis, P. A System Architecture for Smart Farming on Mushroom Cultivation. In Proceedings of the 2022 5th World Symposium on Communication Engineering (WSCE), Nagoya, Japan, 16–18 September 2022; pp. 89–94. [CrossRef]
29. Ferrández-Pastor, F.J.; García-Chamizo, J.M.; Nieto-Hidalgo, M.; Mora-Pascual, J.; Mora-Martínez, J. Developing Ubiquitous Sensor Network Platform Using Internet of Things: Application in Precision Agriculture. *Sensors* **2016**, *16*, 1141. [CrossRef] [PubMed]
30. Carpio, F.; Jukan, A.; Sanchez, A.I.M.; Amla, N.; Kemper, N. Beyond Production Indicators: A Novel Smart Farming Application and System for Animal Welfare. In Proceedings of the Fourth International Conference on Animal-Computer Interaction, Milton Keynes, UK, 21–23 November 2017; pp. 1–11. [CrossRef]
31. Verma, S.; Gala, R.; Madhavan, S.; Burkule, S.; Chauhan, S.; Prakash, C. An Internet of Things (IoT) Architecture for Smart Agriculture. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation, Pune, India, 16–18 August 2018; pp. 1–4. [CrossRef]
32. Zamora-Izquierdo, M.A.; Santa, J.; Martínez, J.A.; Martínez, V.; Skarmeta, A.F. Smart Farming IoT Platform Based on Edge and Cloud Computing. *Biosyst. Eng.* **2019**, *177*, 4–17. [CrossRef]
33. Ray, P.P. Internet of Things for Smart Agriculture: Technologies, Practices and Future Direction. *J. Ambient Intell. Smart Environ.* **2017**, *9*, 395–420. [CrossRef]

34. Khan, F.A.; Abubakar, A.; Mahmoud, M.; Al-Khasawneh, M.A.; Alarood, A.A. Cotton Crop Cultivation Oriented Semantic Framework Based on IoT Smart Farming Application. *Int. J. Eng. Adv. Technol.* **2019**, *8*, 480–484.
35. Ramli, M.I.; Ariffin, M.A.M.; Zainol, Z.; Amin, M.N.M.; Hirawan, D.; Sumitra, I.D.; Jamil, N. Design of a Smart Portable Farming Kit for Indoor Cultivation Using the Raspberry Pi Platform. *Pertanika J. Sci. Technol.* **2023**, *31*, 1731–1754. [CrossRef]
36. Chen, L.; Qian, L.; Zhang, X.; Li, J.; Zhang, Z.; Chen, X. Research Progress on Indoor Environment of Mushroom Factory. *Int. J. Agric. Biol. Eng.* **2022**, *15*, 25–32. [CrossRef]
37. Suresh, M.; Srinivasan, M.; Gowri Shankar, S.; Karthikeyan, D.; Nakhul, V.; Naveen Kumar, A.; Sundar, S.; Maniraj, P. Monitoring and Automatic Control of Various Parameters for Mushroom Farming. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1055*, 012011. [CrossRef]
38. Yin, H.; Yi, W.; Hu, D. Computer Vision and Machine Learning Applied in the Mushroom Industry: A Critical Review. *Comput. Electron. Agric.* **2022**, *198*, 107015. [CrossRef]
39. Kassim, M.R.M.; Harun, A.N.; Yusoff, I.M.; Mat, I.; Kuen, C.P.; Rahmad, N. Applications of Wireless Sensor Networks in Shiitake Mushroom Cultivation. In Proceedings of the 2017 Eleventh International Conference on Sensing Technology, Sydney, NSW, Australia, 4–6 December 2017; pp. 1–6. [CrossRef]
40. Mohammed, M.F.; Azmi, A.; Zakaria, Z.; Tajuddin, M.F.N.; Isa, Z.M.; Azmi, S.A. IoT Based Monitoring and Environment Control System for Indoor Cultivation of Oyster Mushroom. *J. Phys. Conf. Ser.* **2018**, *1019*, 012053. [CrossRef]
41. Dipali, D.; Subramanian, T.; Kumaran, G.S. A Smart Oyster Mushroom Cultivation Using Automatic Fuzzy Logic Controller. *J. Discret. Math. Sci. Cryptogr.* **2023**, *26*, 601–615. [CrossRef]
42. Dong, J.; Zheng, L. Quality Classification of Enoki Mushroom Caps Based on CNN. In Proceedings of the 2019 IEEE 4th International Conference on Image, Vision and Computing, Xiamen, China, 5–7 July 2019; pp. 450–454. [CrossRef]
43. Lu, C.P.; Liaw, J.J. A Novel Image Measurement Algorithm for Common Mushroom Caps Based on Convolutional Neural Network. *Comput. Electron. Agric.* **2020**, *171*, 105336. [CrossRef]
44. Wei, B.; Zhang, Y.; Pu, Y.; Sun, Y.; Zhang, S.; Lin, H.; Zeng, C.; Zhao, Y.; Wang, K.; Chen, Z. Recursive-YOLOv5 Network for Edible Mushroom Detection in Scenes with Vertical Stick Placement. *IEEE Access* **2022**, *10*, 40093–40108. [CrossRef]
45. Moysiadis, V.; Kokkonis, G.; Bibi, S.; Moscholios, I.; Maropoulos, N.; Sarigiannidis, P. Monitoring Mushroom Growth with Machine Learning. *Agriculture* **2023**, *13*, 223. [CrossRef]
46. Aguirre, L.; Frias, J.M.; Barry-Ryan, C.; Grogan, H. Modelling Browning and Brown Spotting of Mushrooms (*Agaricus bisporus*) Stored in Controlled Environmental Conditions Using Image Analysis. *J. Food Eng.* **2009**, *91*, 280–286. [CrossRef]
47. Barauskas, R.; Kriščiūnas, A.; Čalnerytė, D.; Pilipavičius, P.; Fyleris, T.; Daniulaitis, V.; Mikalauskis, R. Approach of AI-Based Automatic Climate Control in White Button Mushroom Growing Hall. *Agriculture* **2022**, *12*, 1921. [CrossRef]
48. Rahman, H.; Faruq, M.O.; Abdul Hai, T.B.; Rahman, W.; Hossain, M.M.; Hasan, M.; Islam, S.; Moinuddin, M.; Islam, M.T.; Azad, M.M. IoT Enabled Mushroom Farm Automation with Machine Learning to Classify Toxic Mushrooms in Bangladesh. *J. Agric. Food Res.* **2022**, *7*, 100267. [CrossRef]
49. Chong, J.L.; Chew, K.W.; Peter, A.P.; Ting, H.Y.; Show, P.L. Internet of Things (IoT)-Based Environmental Monitoring and Control System for Home-Based Mushroom Cultivation. *Biosensors* **2023**, *13*, 98. [CrossRef] [PubMed]
50. Goldstein, M.; Uchida, S. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLoS ONE* **2016**, *11*, e0152173. [CrossRef]
51. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]