agriculture

MDPI

Article

MushR: A Smart, Automated, and Scalable Indoor Harvesting System for Gourmet Mushrooms

Anant Sujatanagarjuna †, Shohreh Kia †, Dominique Fabio Briechle † and Benjamin Leiding *,†

Institute for Software and Systems Engineering, Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany; anant.sujatanagarjuna@tu-clausthal.de (A.S.); shohreh.kia@tu-clausthal.de (S.K.); dominique.fabio.briechle@tu-clausthal.de (D.F.B.)

* Correspondence: benjamin.leiding@tu-clausthal.de

† These authors contributed equally to this work.

Abstract: Gourmet mushrooms are foraged from the wild or grown indoors in controlled environments. Indoor mushroom farms with controlled growth environments allow for all-year-round growing. However, it remains a labor-intensive process. We propose MushR as a modular and scalable gourmet mushroom growing and harvesting system that goes beyond the state of the art, which merely monitors and controls the growing environment, by introducing an image recognition system that determines when and which mushrooms are ready to be harvested in conjunction with a proof of concept of an automated mushroom harvesting mechanism for harvesting the mushrooms without human interaction. The image recognition setup monitors the growing status of the mushrooms and guides the harvesting process. We present a Mask R-CNN model for the detection of oyster mushroom maturity with a 91.7% training accuracy and a semiautomated harvesting system, integrating a Raspberry Pi for control, an electrical switch, an air compressor, and a pneumatic cylinder with a cutting knife to facilitate timely mushroom harvesting. The modularity and scalability of the system allow for industry-level usage and can be scaled according to the required mushroom-growing systems within the facility. The AI model, its underlying dataset, a digital twin for mushroom production, the setup of our growth and control chambers, and additional information are all made available under an open-source license.

Keywords: gourmet mushroom; digital twin; AI; Mask R-CNN; IoT; automation; sustainability

Citation: 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. https://doi.org/10.3390/agriculture13081533

Academic Editor: Gniewko Niedbała, Sebastian Kujawa, Tomasz Wojciechowski and Magdalena Piekutowska

Received: 1 July 2023
Revised: 24 July 2023
Accepted: 27 July 2023
Published: 1 August 2023

1. Introduction

Gourmet mushrooms such as shiitake, oyster, and enoki are harvested in the wild or grown indoors in controlled environments. Harvesting mushrooms outside is a seasonal activity and thus is limited to a few months per year. Moreover, it is a labor-intensive process, and the changing outdoor conditions result in volatile harvests. Furthermore, climate change further limits outdoor mushroom growing and harvesting opportunities [1,2]. Indoor mushroom farms with controlled growth environments allow for an all-year-round growing and harvesting of mushrooms in sensor-controlled grow rooms and grow tents. Additionally, some approaches towards automated production of button mushrooms exist, e.g., [3].

However, those do not apply to gourmet mushrooms, and thus, it remains a labor-intensive process that requires skilled workers [4]. Moreover, gourmet mushroom production often relies on one-time-use plastic bags to hold the substrate. The plastic bag is cut open to start the mushroom fruiting process, which renders the plastic bag useless once the mushrooms have been harvested, thereby creating large amounts of plastic waste. Some production facilities have moved to reusable plastic containers or jars. Whether those are more sustainable remains an open question [5,6].

MushR aims to fill this gap by introducing a modular and scalable gourmet mushroom growing and harvesting system that extends the state of the art—which only monitors

and controls the growing environment—by introducing an image recognition system that detects when and which mushrooms are ready to be harvested in combination with a proof of concept of an automated mushroom harvesting mechanism for harvesting the mushrooms without human interaction. The image recognition setup monitors the growing status of the mushrooms and guides the robot arm during the harvesting process. The modularity and scalability of the system allow for industry-level usage and can be scaled according to the required mushroom-growing systems within the facility. As a result, MushR drastically reduces the necessity of manual labor for gourmet mushroom growing/harvesting and allows for further industrial-scale automation and increased yields and quality of mushrooms.

Specifically, this work addresses the following research questions:

RQ How to automate/digitize/enable sustainable indoor cultivation and harvesting of gourmet mushrooms?

- **RQ1** How to create a digital twin for the gourmet mushroom production process?
- **RQ2** How to monitor gourmet mushroom growth and detect which mushrooms are ready for harvest?
- **RQ3** How to automate harvesting of gourmet mushrooms?

Note that the production of oyster mushrooms (*Pleurotus ostreatus*) serves as a running case of this research as oyster mushrooms are among the most common gourmet mushrooms. Results obtained from the running case can be abstracted to other types of gourmet mushrooms due to their string similarities of the production processes and parameters.

The paper is structured as follows: Section 2 presents supplementary literature, related works, and sustainability in mushroom production. In contrast, Section 3 describes an approach towards a digital twin for (gourmet) mushroom production. Section 4 details an AI model used to monitor the growth stages of gourmet mushrooms and the underlying dataset used to train the model that we created as part of this research. Section 5 outlines automated harvesting mechanisms for gourmet mushrooms. The results of our evaluation are presented in Section 6. Finally, in Section 7, we give our conclusions and outline possible directions for future research.

2. Supplementary Literature and Related Work

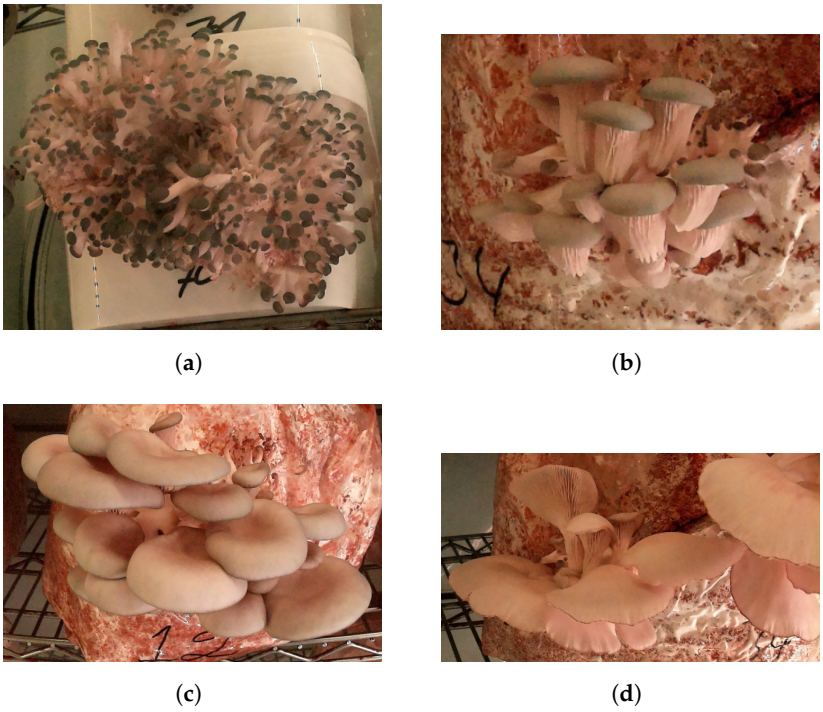
This section provides background information and supplementary literature and introduces related works. Section 2.1 briefly outlines the general process of gourmet mushroom production, while Section 2.2 focuses on related works. Finally, Section 2.3 details a more sustainable approach based on reusable mushroom pods.

2.1. Gourmet Mushroom Production

Indoor mushroom farms allow for an all-year-round growing and harvesting of mushrooms in sensor-controlled grow rooms and grow tents. The mushrooms are grown on substrate blocks, e.g., straw or wood chips, in one-time-use plastic bags. After a prepared substrate block is inoculated with a sample of mycelium (mycelium is a rootlike structure of a fungus consisting of a mass of branching, threadlike hyphae), it is kept in a dark, sterile environment for an incubation period. During incubation, the mycelium completely colonizes the substrate block, after which the plastic bags are cut open and placed in an environment with high humidity to initiate the fruiting of the mushrooms. They are typically harvested in cycles—so-called flushes—with idling periods in between flushes for the mushrooms (or, more precisely, the mushroom mycelium) to recover before triggering the next flush and starting the fruiting of mushrooms again. The substrate is depleted after about three flushes. While fruiting, the mushroom body grows fast and needs to be constantly monitored to be harvested at the right time. Additionally, the growing environment must be in perfect conditions, e.g., humidity, temperature, and air quality.

Figure 1 depicts oyster mushrooms in four different growth stages that we considered for the subsequent training of our AI model: First, in stage I, the fruiting process begins,

and the mushrooms start pinning. Next, in stage II, the mushroom body grows. Third, the mushroom matures and is ready to be harvested. In stage IV, the mushroom has passed the optimal harvesting time window and is now considered overdue.



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

(**c**) (**d**)

Figure 1. Progressing growth stages of *Pleurotus ostreatus*; (**a**) Stage I; (**b**) Stage II; (**c**) Stage III; (**d**) Stage IV.

Learning when and how to harvest mushrooms properly requires training and practice. Especially since harvesting is about more than just selecting the largest mushrooms. On the one hand, they tend to grow in dense clusters, and if they are not adequately thinned out (by hand), the mushrooms might damage each other. Moreover, it prevents the healthy, sustainable growth of the mushroom block, resulting in a decreased yield. Damaged or deformed mushrooms have to be sold at a discount. On the other hand, a particular cluster density is required; otherwise, the yield drops as well. Therefore, even though indoor mushroom growing facilities allow for an all-year-round harvest and optimized yield via controlled growing environments, it remains a labor-intensive process that requires skilled workers [4].

2.2. Related Work

Some level of automation can be achieved for button mushrooms (also known as chestnut mushrooms). Button mushrooms are grown on large soil beds instead of substrate blocks and are subsequently harvested by hand. Ref. [3] presented a robot arm-controlled suction cup for harvesting button mushrooms with a maturity recognition accuracy of 70.93 percent. Moreover, approaches towards sorting button mushrooms have been explored [7]. In contrast to button mushrooms, most gourmet mushrooms, such as the king oyster or the oyster mushroom, do not have a suction-cup-compatible surface. Furthermore, they are more sensitive than button mushrooms and thus require special care in their growing environment and during harvesting. Thus far, they are grown and harvested manually with little process automation; i.e., only the indoor growing environment is controlled and managed using a sensor–actuator setup for variables, such as temperature, humidity, and air quality [8,9]. Some rudimentary (semi-)automated harvesting mechanisms for mushrooms exist, e.g., [10–12], but they do not recognize the mushroom growth

status and optimal harvest time. They do not consider mushroom damage caused by the harvesting process.

Other works focus solely on detecting the growth stage of oyster and button mushrooms using different versions of the YOLO (You Only Look Once) object detection algorithm that have been proposed in the past, e.g., [13–15]. However, neither the model nor the underlying datasets have been published. Moreover, the combination of growth stage detection and automated harvesting was neglected.

Finally, some works consider reusable jars/bottles for mushroom production, e.g., [5,6], and some industry entities started using reusable plastic buckets instead of one-time-use-only mushroom bags [16,17]. However, an analysis of the environmental benefit of using reusable mushroom pods is missing.

2.3. Reusable Mushroom Pods

Aiming to reduce, if not eliminate, the plastic waste created using one-time-use plastic (polypropylene) bags as substrate containers, we experimented with an alternative approach: reusable plastic buckets.

In our experimental setup, we use plastic (polypropylene) buckets, which, weighing at 90 g, can contain up to 3 L of substrate. As shown in Figure 2, we drill 3.5 cm holes on five sides of the bucket, which function as fruiting and ventilation holes. These holes are sealed with a microporous tape during the incubation phase, which is removed for fruiting. Unlike one-time-use plastic bags, the plastic buckets, which we call mushroom pods, are never permanently damaged during the lifetime of the substrate contained therein.

We compute the environmental benefit of using reusable mushroom pods (in terms of materials) over one-time-use plastic bags by running life-cycle impact assessment calculations using OpenLCA [18]. To this end, we define the reference flow and functional unit for both container types in terms of "amount of polypropylene (g)" and "colonizable volume of the container in litres (L)". We further quantify the relationship between the defined reference flow and the functional unit for both container types. Through experimentation, we estimate the colonizable volume of a 5 L one-time-use plastic bag weighing 30 g to be 3 L (on average). This decrease is because the bags do not have a built-in method to seal the contained substrate. A substantial amount of the bag is used for this purpose, accomplished by folding the opening over itself several times. Unlike the bags, our mushroom pods are equipped with a sealable lid. As a consequence, the estimated colonizable volume of our 3 L mushroom pods is still 3 L.

Using these calculations, we create a reference process for both these container types in OpenLCA. We source the life-cycle inventory (LCI) data pertaining to the manufacturing processes of polypropylene bags and buckets from the Agribalyse dataset [19]. The following are the life-cycle impact assessment (LCIA) results computed using the BEES+ impact assessment method:

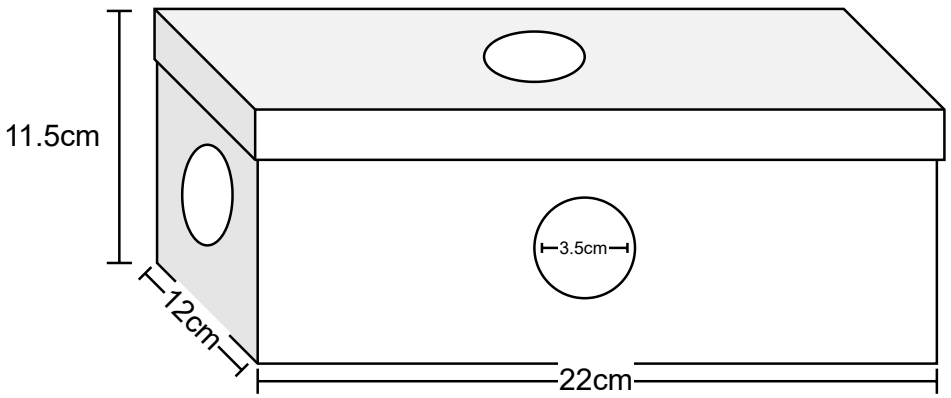


Figure 2. Graphical representation of the MushR mushroom pods.

From Table 1, we can infer the benefit of our reusable mushroom substrate pods over the non-reusable substrate bags with respect to the impact categories. Using "global warming" as an example, a 5 L (3 L usable) one-time-use bag has an impact of 102.46 g CO_2 eq, compared with a 3 L reusable pod with 217.20 g CO_2 eq. From this, we can infer that one reusable pod will have less impact in this category than three non-reusable bags. This means that simply reusing our mushroom pods three times will result in them having a lower impact in this category than the non-reusable counterparts. Similar comparisons can be made with other impact categories. The complete LCIA calculations and results are given in Appendix A. The above figures indicate that the reusable mushroom pods will have a lower environmental impact over time. These figures, however, are highly dependent on the specific materials used for producing the containers. These figures also might not scale linearly with the container size for mass-producing mushrooms.

Table 1. LCIA results comparison of 3 L colonizable volume of one-time-use bags and reusable pods.

Impact Category	Reference Unit	One-Time-Use Bag	MushR Reusable Pods
Acidification	H+ mmole eq	11.21	40.15
Ecotoxicity	g 2,4-D eq	2.07	0.72
Eutrophication	g N eq	0.43	0.34
Global warming	g CO_2 eq	102.46	217.20
Habitat alteration	T&E count	1.88×10^{16}	-7.03×10^{17}
HH cancer	g C_6H_6 eq	0.15	0.48
HH criteria air pollutants	microDALYs	0.004	0.015
HH noncancer	g C_7H_7 eq	423.69	1124.07
Indoor air quality	g TVOC eq	0	0
Natural resource depletion	MJ surplus	0.26	0.92
Ozone depletion	g CFC-11 eq	2.64×10^{7}	1.49×10^{6}
Smog	g NO_x eq	0.14	0.47
Water intake	liters	0.95	2.3

3. Digital Twins for Gourmet Mushrooms

In the food industry, digital twins can provide better food quality, predictive maintenance, energy-use minimization, and higher transparency of the production processes [20].

The digital twin implementation developed for this project models the state of the key assets throughout various stages of the mushroom production process. Since the state of these assets is tightly linked with the state(s) of other inter-related assets, we chose to model this information using a graph database, Neo4j [21]. The various assets are modeled as nodes, and their temporal and nontemporal relationships are modeled using relations.

Figure 3 shows a class diagram of the graph database schema created for the database. Figure 3a shows a class hierarchy of the various node classes stored in the database. It further shows the possibility of relations that can be created between two node classes, including the directionality. These relations themselves have a class hierarchy, shown in Figure 3b.

Neo4j is incapable of enforcing rigid class hierarchy or data types of their attributes of nodes or relations. For this reason, we implement the class hierarchy using the Neomodel Object Graph Mapper (OGM) for Neo4j, which is then exposed using a REST-API, the MushR DigitalTwin API. All interactions with the data, including the creation of new assets, are handled through this API. Since the API itself does not store any of the digital twin data, the scalability of Neo4j's various enterprise solutions can be leveraged to handle increasing amounts of data without compromising the API's performance.

The defined schema allows the retrieval of the state of the network at any previous instance in time. For example, the graph stores the state of a substrate container and contains information on where it is located, which can be either a storage location or a grow chamber. The relation between an instance of a substrate with its substrate container implicitly gives its location history. Substrates are inoculated using samples of an existing mycelium spawn or newly procured mushroom strains.

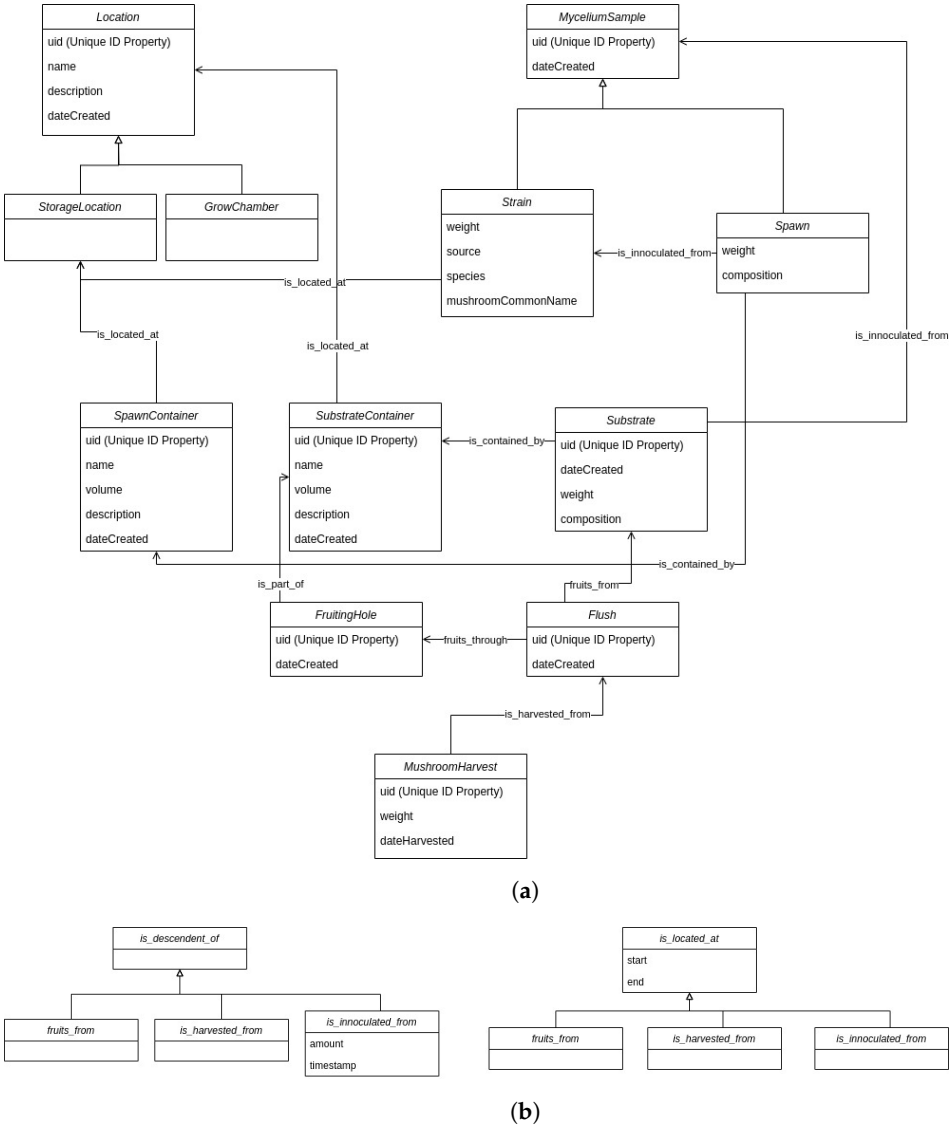


Figure 3. Digital twin database class diagram; (**a**) Node Classes; (**b**) Relation Classes.

The provenance of a specific mushroom harvest can be traced using the digital twin by recursively traversing the relations. Figure 4 shows a graph visualization of provenance based on data recorded over 8 months in 2022. The data are restricted to 300 nodes and relations pertaining to inoculation, fruiting, and harvests. The digital twin database also stores location-related information so that the location of any of the assets can be accessed at any instance of time.

More information about the MushR digital twin implementation can be found in our GitHub repository (Appendix A).

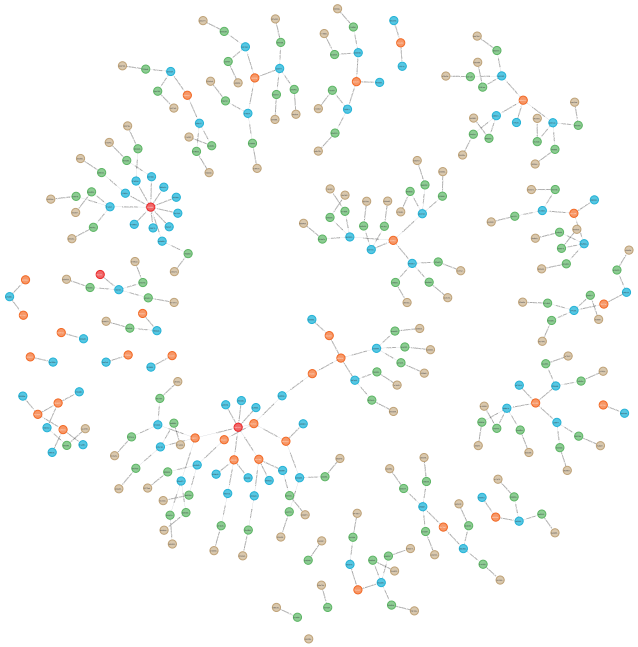


Figure 4. Digital twin graph visualization showing provenance of mushrooms (brown) harvested from flushes (green) fruiting from substrate (blue) inoculated from spawn (orange) or strain (red).

4. Detection and Monitoring of Gourmet Mushroom Growth Stages

Subsequent Section 4.1 provides details on our open-source dataset, followed by Section 4.2, which focuses on training our AI model and its predictions for gourmet mushroom growth stages.

4.1. Dataset

The dataset created for this project focuses on capturing images of the mushroom-growing environment from three different perspectives within each of our two growth tents for mushroom production. Instead of providing images of every individual bucket and mushroom, we capture the overall scene and its variations. The images from each perspective are captured simultaneously and automatically hourly. This approach allows for monitoring the development and maturity of the oyster mushrooms over time. We captured and accumulated 34,400 images using six Raspberry Pi HQ Cameras [22] equipped with wide-angle lenses (120° vertical field of view) over a period of 10 months to create a comprehensive dataset.

In the dataset preparation phase, we clean data to ensure that the focus of our project, which is detecting the maturity of oyster mushrooms, is maintained. As part of this process, we remove images that are foggy and have a significant amount of noisy areas or are contaminated. The selected dataset only included images where the mushrooms were clearly visible, allowing for training the model specifically for maturity detection. However, it is important to note that we have preserved the original raw dataset, including the foggy and black images on Kaggle (Appendix A). This is performed to provide a comprehensive dataset for other developers working on related projects or requiring access to diverse images. Researchers and enthusiasts in the field of agriculture automation can access and utilize the dataset for further analysis and experimentation. Since the images are not only from buckets and mushrooms, they enable the development of new algorithms and approaches for mushroom maturity detection and other related works. In addition to the raw images, we provide annotations for a subset of the dataset. Annotations were carefully created to mark the regions of mushrooms within the images. The annotation approach involves manual labeling of mushroom maturity regions based on visual cues, such as the mushrooms' color, shape, and size. The annotation provides a valuable resource for training and validating the Mask R-CNN model.

4.2. Training and Prediction

The machine learning model employed for the maturity detection of oyster mushrooms in this project is Mask R-CNN (region-based convolutional neural network) [23]. The model belongs to the family of instance segmentation models, which detect objects within an image and delineate their precise boundaries through pixel-level segmentation. This model is well suited for our task as it enables the accurate identification and localization of mature mushrooms within the captured images. The Mask R-CNN model architecture combines two key components, a region proposal network (RPN) [24] and a fully convolutional network (FCN) [25]. RPN generates a set of potential object proposals, while FCN performs pixelwise segmentation and classification for each proposed region. This two-stage approach allows the model to achieve robust performance by leveraging local and global contextual information effectively. To train this model, we annotated a subset of the custom dataset using CVAT [26]. We annotated *individual* mushroom fruiting bodies into three classes (based on three of the four stages outlined in Section 2.1): "not-ready", "ready", and "overdue", indicating whether the mushroom can be harvested. The annotated subset of images serves as the ground truth for training and evaluating the model.

To train the model, we rely on the Detectron2 [27] library, a popular computer vision framework, for instance, segmentation that provides a comprehensive set of tools and predefined architectures for object detection and instance segmentation tasks. In our research, the Detectron2 library is the foundation for training and fine-tuning the Mask R-CNN model on our custom dataset. It optimizes the model's performance specifically for oyster mushroom maturity detection. The Mask R-CNN model, integrated with the Detectron2 library, proves to be a robust and effective solution for the maturity detection of oyster mushrooms. See Table 2.

Table 2. MushR dataset overview, number of annotated instances per class.

Dataset	Not-Ready	Ready	Overdue	Total
Train	723	1344	692	2759
Test	251	198	92	541

For the training of the model, we initialize the model with weights pretrained from the COCO dataset [28]. We then train the model on our custom training dataset. We adopted the default Detectron2 hyperparameters (https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml (accessed on 30 June 2023)) and used a base learning rate (post-warm-up) of 0.00015, with a batch size of 2 for 40,000 iterations. The model trained on an Nvidia GTX 4090 using 2.7 GB of memory at 10 iterations per second, and achieved a final training accuracy of 91.72%, as shown in Figure 5.

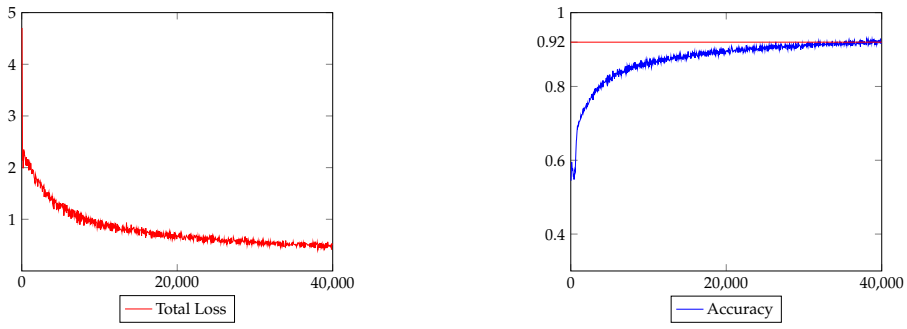
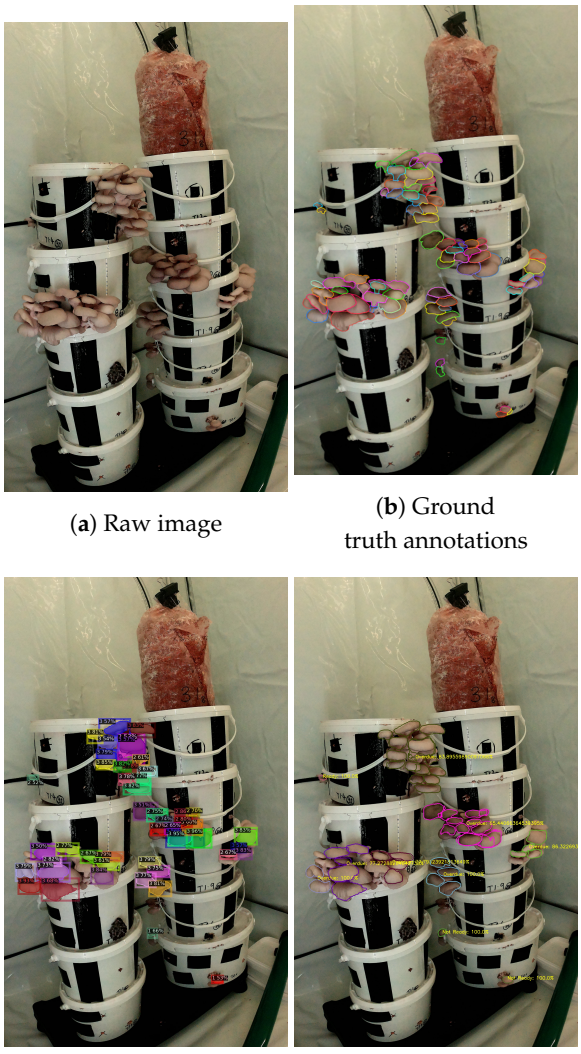


Figure 5. Mask R-CNN training curve.

Although the trained Mask R-CNN model segments and classifies *individual* mushrooms, in mushroom production, entire flushes are always harvested together rather than individual mushrooms. Therefore, an additional step is required to cluster the predicted masks and obtain a final prediction for the entire flush. This clustering process helps to consolidate the individual mushroom predictions into a cohesive prediction for the entire harvest, ensuring efficient and accurate harvesting in mushroom production.

We use the DBSCAN algorithm to cluster the predicted masks, computing the center of the masks to be used as proxy. We use $\epsilon = 2.5 \times 2 \times r$, where r is the average (approximate) radius of all the mask instance predictions in the input image. This value defines the neighborhood of a mask center point, and any masks that do not center within this neighborhood are not considered part of the same flush. Figure 6 shows example results of our maturity prediction workflow performed on a single image. In addition to this clustering, we compute each cluster's total area of the instance classes, Not-Ready, Ready, and Overdue. This computation improves the accuracy of the final maturity state prediction for the cluster over using a simple instance count.



(**a**) Raw image (**b**) Ground truth annotations

(**c**) Mask-RCNN inference (**d**) Final clustering results

Figure 6. Example of oyster mushroom maturity prediction on a single image showing (**a**) raw image used for training; (**b**) annotated ground truth instances; (**c**) Mask R-CNN inference results on trained model, with each instance labeled with the predicted class; Not-Ready(1)/Ready(2)/Overdue(3) and class probability (%); and (**d**) final harvesting decision by clustering predicted instance masks, including class probability (%).

More information, such as the complete training configuration, train and test datasets used, and code used for clustering and visualization can be found in our GitHub repository (Appendix A).

5. Harvesting Gourmet Mushrooms

As outlined in Sections 2.1 and 2.2, it is infeasible to use suction cups to harvest gourmet mushrooms. In pursuit of the goal of a prototype harvesting system for gourmet, we followed two possible approaches: pneumatic and motorized harvesting, detailed in Sections 5.1 and 5.2, respectively.

5.1. Pneumatic Harvesting

Figure 7 shows the components used to prototype the pneumatic harvesting system. This system actuates a blade (B), which is mounted perpendicularly on a bidirectional piston rod cylinder (C). Figure 8 shows how the cylinder is set up to harvest oyster mushrooms growing from our developed mushroom pods. The pneumatic tubes T_1 and T_2 control the motion of the pneumatic cylinder. Depending on whether T_1 or T_2 is pressurized, the piston rod moves outwards or inwards from the cylinder, respectively. When both T_1 and T_2 are pressurized or depressurized, the piston is immobilized or free to move, respectively. For the purpose of harvesting, T_1 and T_2 must be pressurized and depressurized alternatively. This is accomplished by the 5/3-way solenoid valve (SV). Internally, two solenoids control the valves pressurizing T_1 and T_2. These solenoids are activated by 12-volt DC magnetic coils, MG_1 and MG_2. These coils are digitally controlled by 5-volt relay switches connected to a Raspberry Pi 4B [29], programmatically controlling the entire setup. More information about our experimental setup can be found in Appendix A.

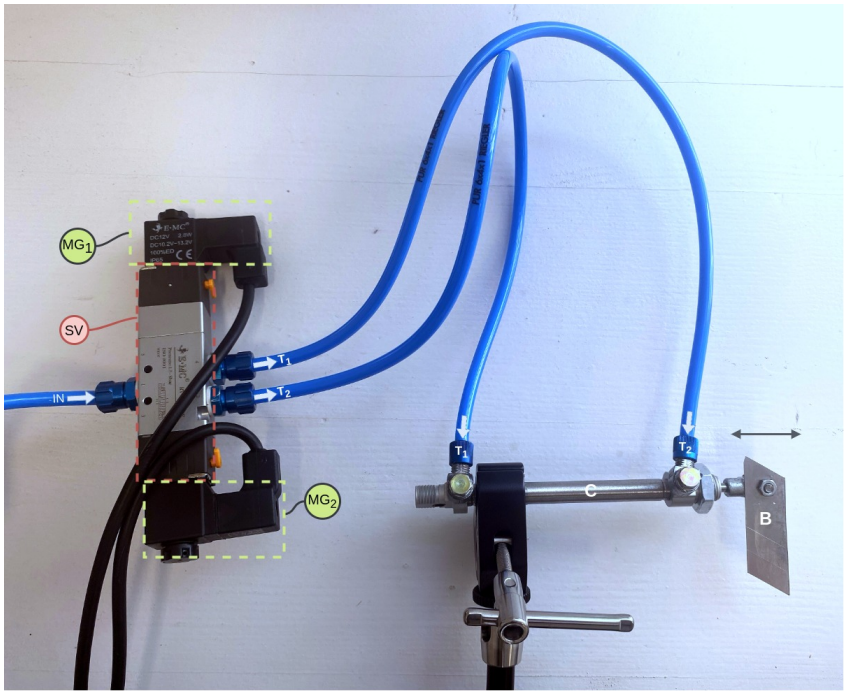


Figure 7. Pneumatic harvesting system components.

Figure 8. Pneumatic harvesting system setup to harvest mushrooms from a MushR mushroom pod.

5.2. Motorized Harvesting

Figure 9 shows the major components of the developed motorized harvesting system, which consists of a stainless steel cylinder (H) with fruiting holes suspended from the ceiling of a stool constructed using item profiles. The cylinder contains a plastic bag filled with a substrate for oyster mushroom growth. These holes on the cylinder align with the perforation in the plastic bag, allowing the mushrooms to emerge and grow. A 3D-printed lid (L), matching the diameter of the cylinder, seals the bottom to prevent substrate spillage in case of damage to the bag. A stepper motor (S) is installed underneath the suspended cylinder to drive a 3D-printed plate, which vertically holds the blade (B). The plate is specifically designed to accommodate the motor, ensuring stability during rotation.

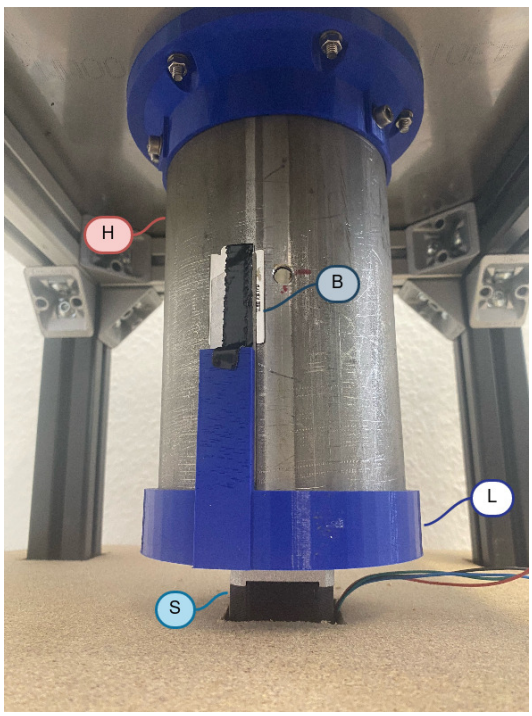


Figure 9. Motorized harvesting system.

The stepper motor initiates rotation, causing the blade to cut the mushrooms protruding through the holes in the cylinder. We program the motor to rotate in either direction to ensure successful harvesting. Adjustments, such as sharpening the knife and modifying the motor's speed, can be made to optimize the system's performance.

6. Evaluation

The developed digital twin implementation models the state of the key assets throughout various stages of the gourmet mushroom production process. While it does not provide process optimization for the gourmet mushroom production process, it is modeled to be capable of storing the temporal state of any of the essential assets, such as substrate, spawn and harvests involved in the production process, as detailed in Section 3.

Tables 3 and 4 evaluate the Mask R-CNN model described in Section 4. We evaluated the trained model in terms of average precision (AP) on our test set using the COCO evaluation metric (https://cocodataset.org/#detection-eval (accessed on 30 June 2023)). Table 5 provides a brief description of the evaluation metrics used.

Table 3. Mask R-CNN model evaluation using the COCO evaluation metric [30].

Criteria	AP	$AP^{IOU=0.50}$	$AP^{IOU=0.75}$	AP^{small}	AP^{medium}	AP^{large}
Bounding Box	61.876	71.016	67.920	20.792	69.788	87.207
Segmentation	49.332	70.639	60.812	9.010	54.799	80.218

Table 4. Mask R-CNN model evaluation (per class).

Task	Class	AP
Bounding Box	Not-Ready	34.386
	Ready	74.683
	Overdue	76.558
Segmentation	Not-Ready	26.502
	Ready	61.083
	Overdue	60.410

As represented in Table 4, the AP of the bounding boxes predicted by the Mask R-CNN model is generally larger than those of segmentation. The "Not-Ready" class of instances also has lower AP than the other classes. A similar trend can be seen in Table 3 with AP across scales: AP^{small} is significantly lower than AP^{medium} and AP^{large}. Since mushroom fruiting bodies that are "Not-Ready" are usually smaller in size, the resolution of the training images seems to play a role in the decrease in AP. The model has its largest AP under the evaluation criterion AP^{large}.

Table 5. Description of COCO metrics (adapted from [30]).

Metric	Description
AP	AP at IoU = 0.50:0.05:0.95
AP^{IOU} = 0.50	AP at IoU = 0.50 (PASCAL VOC metric)
AP^{IOU} = 0.75	AP at IoU = 0.75 (strict metric)
AP^{small}	AP for small objects: pixel area < 32^2
AP^{medium}	AP for medium objects: 32^2 < pixel area < 96^2
AP^{large}	AP for large objects: area > 96^2

Sections 5.1 and 5.2 describe our prototype pneumatic and motorized harvesting systems, respectively. While they are capable of harvesting full flushes of mushrooms, the time required to harvest a single flush increases heavily on the water content and thickness of the flushes at the base of the fruiting holes. From our limited testing, the pneumatic harvesting mechanism setup, as shown in Figure 8, failed to produce any significant results

below 2.1 bar of pressure. At 7.5 bar, we could reliably harvest flushes, albeit requiring up to 5 s of continuous application to successfully harvest the thickest fruiting flushes. On the other hand, since the fruiting holes in the motorized harvesting system are significantly smaller, fruiting flushes can be harvested nearly instantaneously, given that the blade used is sufficiently sharp. The downside of using a smaller fruiting hole, however, is a decrease in the fruiting speed and yield quantity of the flushes.

Unlike the motorized harvesting mechanism, which uses a dedicated harvesting mechanism for each substrate container, the pneumatic harvesting system is an independent modular component. In an industrial-grade setup, since our mushroom pods have flat surfaces, they can be moved from their growth chambers when the mushrooms are ready to harvest (as detected by our Mask R-CNN-based maturity detection method) to the harvesting system via conveyor belts (or other industrial mobility solutions). The number of mushroom pods being used in production can hence increase faster than the number of harvesting systems, the rate of the increase being dependent on the desired throughput of the production environment, thereby allowing the system to be very scalable.

Discussion and Limitations

Our three main contributions, the digital twin, the Mask R-CNN-based maturity detection, and the automated harvesting mechanisms, are novel in their application towards gourmet mushroom production. They do, however, have several limitations that require further research to amend.

The digital twin implementation described in Section 3 did not use any process modeling to monitor or simulate the various production processes involved in gourmet mushroom production.

While our Mask R-CNN model's performance is promising, there are certain limitations to consider. Factors such as lighting conditions, oyster mushroom variations, and occlusions may affect the model's accuracy. As mentioned in Section 4, we accumulated 34,400 images of oyster mushrooms. However, due to the immensely time-consuming endeavor of annotating the mushrooms with masks, we could only annotate a small portion of those images. This subsequently limits the potential of our Mask R-CNN model to learn to segment these mushrooms.

Lastly, our developed methods have yet to be tested in an industrial gourmet mushroom production environment, and the results presented in this paper have the possibility of being only applicable to our experimental setup for oyster mushroom production.

7. Conclusions and Future Work

This work makes four contributions: First, we present a digital twin for gourmet mushroom production, which is capable of storing the temporal state of any of the important assets, such as substrate, spawn and harvests involved in the mushroom production process. We also provide a visual user interface to interact with the graph-based structure of the digital twin representation. Second, we introduce an image recognition system that determines when and which mushrooms are ready to be harvested in conjunction with a proof-of-concept of an automated mushroom harvesting mechanism for harvesting the mushrooms without human interaction. The image recognition setup monitors the growing status of the mushrooms and guides the harvesting process. Third, we present a Mask R-CNN model for the detection of oyster mushroom maturity, which has an evaluated average precision of up to 80.2 (AP^{large}), as well as a semiautomated harvesting system that can be scaled, integrating a Raspberry Pi for control, an electrical switch, an air compressor, and a pneumatic cylinder with a cutting knife to facilitate timely mushroom harvesting as a third contribution. Fourth, we perform an analysis of the environmental benefit of using reusable mushroom pods in favor of one-time-use-only plastic bags. In our use case, reusing the mushroom pods three times yields a more eco-friendly output than previous approaches relying on plastic bags.

Finally, the AI model, its underlying dataset, a digital twin for mushroom production, the setup of our growth and control chambers, and additional information are all made available under an open-source license.

Future improvements could involve collecting a more extensive and more diverse dataset to enhance the model's robustness further, e.g., different types of gourmet mushrooms but also different settings. Additionally, fine-tuning the model with additional training iterations or exploring alternative architectures may yield even better results. Besides the automated harvesting mechanism, subsequent research, i.e., future work, may optimize the growth environment parameters to maximize mushroom growth and harvest based on collected sensor data and the digital twin/shadow data of previously harvested mushroom pods.

While this work focused on monitoring and detecting gourmet mushroom growth stages and subsequent harvesting activities, future work may also focus on other issues of the mushroom production process. Recognizing contaminated substrate or mushrooms further reduces human intervention. Our open-source dataset includes images exhibiting contamination, where mushrooms, buckets, tents, or even the camera may be contaminated. By keeping the complete dataset, we aim to support future research and enable the development of new projects beyond the scope of this specific maturity detection project.

Moreover, the monitoring system could also be coupled with the control unit of the growth environment to manipulate growth-related parameters (e.g., temperature) so that mushrooms are ready to be harvested on predetermined days.

Beyond the production of gourmet mushrooms, the solutions, e.g., growth/harvest monitoring and detection and the robot-based harvesting developed in MushR, may be generalized to other indoor farming activities in controlled environments beyond mushrooms and, in special cases, even for outdoor farming activities.

Author Contributions: Conceptualization, A.S., S.K., D.F.B. and B.L.; Methodology, A.S., S.K., D.F.B. and B.L.; Software, A.S. and S.K.; Validation, A.S. and S.K.; Investigation, S.K., D.F.B. and B.L.; Resources, A.S.; Data curation, A.S. and S.K.; Writing—original draft, A.S., S.K. and B.L.; Writing—review & editing, A.S., D.F.B. and B.L.; Visualization, A.S.; Supervision, B.L.; Project administration, A.S. and B.L.; Funding acquisition, A.S., D.F.B. and B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Cascade Funding via the VEDLIoT Open Call. The VEDLIoT project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 957197. We acknowledge the financial support from the Open Access Publishing Fund of Clausthal University of Technology.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: All data related to this project have been made available to the public and are listed in Appendix A.

Acknowledgments: The authors thank Harish Gundelli, Athira Mavoomkuttathil Sivachandran, Sepideh Sayadkouh, and Johannes Mayer for their support and contribution to the research presented in this work.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

1. MushR GitHub Repository: https://github.com/ETCE-LAB/MushR (accessed on 30 June 2023)
2. AI model and dataset to detect the growth stages of gourmet mushrooms (specifically oyster mushrooms): https://github.com/ETCE-LAB/mushr-mask-r-cnn/ (accessed on 30 June 2023)
3. Raw underlying dataset for the AI model: https://www.kaggle.com/datasets/etcelab/mushr-project-raw-image-dataset-oyster-mushrooms (accessed on 30 June 2023)

4. Github repository of the digital twin for gourmet mushroom production: https://github.com/ETCE-LAB/mushr-digitaltwin-api (accessed on 30 June 2023)
5. Github repository for the growth chamber setup and control environment: https://github.com/ETCE-LAB/MushR/blob/main/growchamber-setup (accessed on 30 June 2023)
6. LCA results (one-time-use plastic bags & MushR reusable mushroom pods): https://github.com/ETCE-LAB/MushR/blob/main/lca-calculations (accessed on 30 June 2023)

References

1. 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] [PubMed]
2. Yang, X.; Luedeling, E.; Chen, G.; Hyde, K.D.; Yang, Y.; Zhou, D.; Xu, J.; Yang, Y. Climate Change Effects Fruiting of the Prize Matsutake Mushroom in China. *Fungal Divers.* **2012**, *56*, 189–198. [CrossRef]
3. Huang, M.; Jiang, X.; He, L.; Choi, D.; Pecchia, J.; Li, Y. Development of a Robotic Harvesting Mechanism for Button Mushrooms. *Trans. ASABE* **2021**, *64*, 565–575. [CrossRef]
4. Huang, M.; He, L.; Choi, D.; Pecchia, J.; Li, Y. Picking Dynamic Analysis for Robotic Harvesting of Agaricus Bisporus Mushrooms. *Comput. Electron. Agric.* **2021**, *185*, 106145. [CrossRef]
5. Mamiro, D.P.; Mamiro, P.S.; Mwatawala, M.W. Oyster Mushroom (*Pleurotus* spp.) Cultivation Technique Using Re-Usable Substrate Containers and Comparison of Mineral Contents With Common Leafy Vegetables. *J. Appl. Biosci.* **2014**, *80*, 7071–7080. [CrossRef]
6. Sánchez, C. Cultivation of *Pleurotus ostreatus* and Other Edible Mushrooms. *Appl. Microbiol. Biotechnol.* **2010**, *85*, 1321–1337. [CrossRef] [PubMed]
7. Wang, F.; Zheng, J.; Tian, X.; Wang, J.; Niu, L.; Feng, W. An Automatic Sorting System for Fresh White Button Mushrooms Based on Image Processing. *Comput. Electron. Agric.* **2018**, *151*, 416–425. [CrossRef]
8. Mohammed, M.; Azmi, A.; Zakaria, Z.; Tajuddin, M.; Isa, Z.; Azmi, S. IoT-based Monitoring and Environment Control System for Indoor Cultivation of Oyster Mushroom. *J. Phys. Conf. Ser.* **2018**, *1019*, 012053. [CrossRef]
9. Ariffin, M.A.M.; Ramli, M.I.; Amin, M.N.M.; Ismail, M.; Zainol, Z.; Ahmad, N.D.; Jamil, N. Automatic Climate Control for Mushroom Cultivation Using IoT Approach. In Proceedings of the 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), Shah Alam, Malaysia, 9 November 2020; pp. 123–128.
10. Rong, J.; Wang, P.; Yang, Q.; Huang, F. A Field-tested Harvesting Robot for Oyster Mushroom in Greenhouse. *Agronomy* **2021**, *11*, 1210. [CrossRef]
11. Yaong, Q.; Rong, J.; Wang, P.; Yang, Z.; Genc, P. Real-time Detection and Localization Using SSD Method for Oyster Mushroom Picking Robot. In Proceedings of the 2020 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Asahikawa, Japan, 28–29 September 2020; pp. 158–163.
12. Huang, M.; Jiang, X.; He, L.; Choi, D.; Pecchia, J. Hand-picking Dynamic Analysis for Robotic Agaricus Mushroom Harvesting. In Proceedings of the 2020 ASABE Annual International Virtual Meeting, Virtual, 13–15 July 2020; p. 1.
13. Lu, C.P.; Liaw, J.J.; Wu, T.C.; Hung, T.F. Development of a Mushroom Growth Measurement System Applying Deep Learning for Image Recognition. *Agronomy* **2019**, *9*, 32. [CrossRef]
14. Moysiadis, V.; Kokkonis, G.; Bibi, S.; Moscholios, I.; Maropoulos, N.; Sarigiannidis, P. Monitoring Mushroom Growth with Machine Learning. *Agriculture* **2023**, *13*, 223. [CrossRef]
15. Wang, Y.; Yang, L.; Chen, H.; Hussain, A.; Ma, C.; Al-gabri, M. Mushroom-YOLO: A Deep Learning Algorithm for Mushroom Growth Recognition Based on Improved YOLOv5 in Agriculture 4.0. In Proceedings of the 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), Perth, WA, Australia, 25–28 July 2022; IEEE: New York, NY, USA, 2022; pp. 239–244.
16. Broussard, W. How to Grow Mushrooms in Buckets & Containers. 2023. Available online: https://northspore.com/blogs/the-black-trumpet/growing-mushrooms-in-buckets-containers (accessed on 30 June 2023).
17. Shields, T. Grow Mushrooms Easy in a 5 Gallon Bucket—Freshcap. 2023. Available online: https://learn.freshcap.com/growing/bucket-grow/ (accessed on 30 June 2023).
18. GreenDelta. OpenLCA—Github Repository. 2023. Available online: https://github.com/GreenDelta/olca-app (accessed on 30 June 2023).
19. AGRIBYLASE Program. Agribylase Dataset 3.0. 2022. Available online: https://nexus.openlca.org/database/Agribalyse (accessed on 30 June 2023).
20. Verboven, P.; Defraeye, T.; Datta, A.K.; Nicolai, B. Digital Twins of Food Process Operations: The Next Step for Food Process Models? *Curr. Opin. Food Sci.* **2020**, *35*, 79–87. [CrossRef]
21. Neo4j. Neo4j: Graphs for Everyone—Github Repository. 2023. Available online: https://github.com/neo4j/neo4j (accessed on 30 June 2023).
22. Raspberry Pi Foundation. Raspberry Pi HQ Camera. 2023. Available online: https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/ (accessed on 18 July 2023).

23. Matterport, Inc. Mask R-CNN for Object Detection and Segmentation—Github Repository. 2019. Available online: https://github.com/matterport/Mask_RCNN (accessed on 30 June 2023).
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
25. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
26. OpenCV. Computer Vision Annotation Tool (CVAT)—Github Repository. 2023. Available online: https://github.com/opencv/cvat (accessed on 30 June 2023).
27. Meta Research. Detection2—Github Repository. 2023. Available online:https://github.com/facebookresearch/detectron2 (accessed on 30 June 2023).
28. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
29. Raspberry Pi Foundation. Raspberry Pi 4 Tech Specs. 2023. Available online: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/ (accessed on 30 June 2023).
30. COCO Consortium. COCO Dataset—Detection Evaluation. 2021. Available online: https://cocodataset.org/#detection-eval (accessed on 30 June 2023).