Duman, S., Elewi, A., Avaroglu, E. (2026). A comprehensive multisensory learning approach for oyster mushroom maturity detection in greenhouses. Agriculture and Forestry, 72 (1): 203-220. https://doi:10.17707/AgricultForest.72.1.12

DOI: 10.17707/AgricultForest.72.1.12

Sonay DUMAN*[1,2], Abdullah ELEWİ[2], Erdinç AVAROĞLU[2]

A COMPREHENSIVE MULTISENSORY LEARNING APPROACH FOR OYSTER MUSHROOM MATURITY DETECTION IN GREENHOUSES

SUMMARY

In the mushroom industry, analysing growth and detecting maturity during cultivation is crucial for smart mushroom farming. This study introduces a comprehensive multisensory approach employing diverse machine learning algorithms to automate the classification of oyster mushroom (*Pleurotus ostreatus*) maturity classes in a greenhouse setting. Three unique datasets were created, comprising manually extracted features of oyster mushroom node sizes (height and width) alongside size features derived from bounding boxes automatically identified by trained Detectron2 and YOLOv8 object detectors utilising annotated mushroom images. The manual and automatic datasets were used to train classification algorithms separately, their performance was evaluated and compared. The impact of environmental setting within the mushroom greenhouse on mushroom growth was examined by integrating environmental variables as features for classification algorithms. The results indicated a significant enhancement in classification accuracy, especially incorporating the temperature and humidity values. The findings indicate that the HistGBDT classifier achieved the highest testing accuracies of 91.2%, 90.1%, and 92.7% on the automatic YOLOv8, Detectron2, and manual datasets, respectively, employing extracted mushroom node’s width and height, along with temperature and humidity features in the greenhouse. The results of other classifiers demonstrated minor differences between manual and automatic datasets, suggesting that the YOLOv8 object detector generated mushroom bounding boxes significantly more aligned with those drawn by humans compared to the Detectron2 framework. These results demonstrate an invaluable multisensory approach for detecting mushroom maturity classes in greenhouse settings, surpassing existing camera-based systems.

Keywords: Oyster mushroom, YOLOv8, Detectron2, agriculture, classification, object detection

---

[1]Sonay Duman (corresponding author: sonay.duman@toros.edu.tr), Toros University, Department of Software Engineering, Mersin, 33140, TURKEY

[2]Sonay Duman, Abdullah Elewi, Erdinç Avaroğlu, Mersin University, Department of Computer Engineering, Mersin, 33140, TURKEY

Notes: The authors declare that they have no conflicts of interest. Authorship Form signed online.

Recieved:09/01/2026 *Accepted:24/02/2026*

INTRODUCTION

Mushroom cultivation has become a critical component of contemporary agriculture due to its numerous economic and environmental benefits, which contribute to the sustainability and diversity of the agricultural industry (Vinci *et al.,* 2023, Jayaraman *et al.,* 2024, Ben Mansour *et al.,* 2025). Smart mushroom cultivation integrates sensing, automation, and data-driven analysis to support controlled growth monitoring and decision-making processes (Elewi *et al.*, 2024, Sangeeta *et al.*, 2024, Badoni and Siddiqui, 2025). Recent progress in indoor environmental studies of mushroom cultivation facilities has focused on the development of environmental monitoring and control systems, the application of computer modelling to address indoor environmental issues, and the refinement of mushroom facility design (Chen *et al.*, 2022). These advancements have led to the development of smart farming systems that enable real-time recording of environmental parameters and automatic regulation of equipment, as well as the establishment of growth models for specific mushroom species. One of the key advantages of smart mushroom cultivation is the ability to control and optimise various parameters that affect mushroom growth and yield (Moysiadis *et al.*, 2023). Parameters such as temperature, relative humidity, and air quality can be remotely monitored and maintained at optimal levels using state-of-the-art technologies (Elewi *et al.*, 2024). This level of control and optimization is not possible in conventional methods of mushroom cultivation, leading to lower yields (Rahman *et al.*, 2022). By implementing smart farming techniques, farmers can improve the yield of mushrooms and achieve higher productivity. In addition to improving productivity, smart mushroom cultivation also offers benefits in terms of standardization and safety. This is particularly important for the commercialization of mushroom cultivation, as it allows consistent quality and safety standards. Furthermore, smart mushroom farming has the potential to transform the mushroom industry into a high-technology industry with high levels of mechanization and automation (Raut, 2019; Yin *et al.,* 2022).

Several studies have investigated different aspects of smart mushroom farming, which includes the application of Internet of Things (IoT), computer vision (CV), artificial intelligence (AI), machine learning (ML), and automation technologies to improve efficiency and decision-making processes. A recent investigation (Yin *et al.*, 2022) looked into how cutting-edge technologies are being used in the mushroom sector. It looked at several CV and ML technologies used for growing, collecting, and grading mushrooms. Using machine learning algorithms to classify mushrooms is a data-driven way that makes the process more accurate while minimising human work (Elewi *et al.*, 2024, Adebayo *et al.*, 2025, Singh *et al.*, 2025). Researchers have looked into using machine learning algorithms for a number of tasks, including finding mushrooms and measuring mushroom caps (Lu and Liaw, 2020), classifying the quality of mushroom spawn (Tongcham *et al.*, 2020), classifying the quality of shiitake mushrooms (Zhu *et al.,* 2023), and identifying the difference between poisonous and edible mushrooms (Rahman *et al.,* 2022). Paudel and Bhatta (2022) employed random forest and representative (REP) tree classifiers for the classification of oyster mushroom growth.

Few studies in the literature have addressed the growth analysis and maturity detection of mushrooms. Lu *et al.* (2019) developed an IoT-based mushroom measurement system utilising YOLOv3 to identify button mushrooms in a controlled experimental setting, with the objective of quantifying mushrooms and estimating their cap sizes and optimal harvest times. Wang *et al.* (2022) conducted a similar study that created a version of YOLOv5 to accurately find and locate shiitake mushrooms using a web-based app. However, this study did not observe mushrooms growth or maturity. Moysiadis *et al.* (2023) conducted one of the most relevant research studies. The authors monitored the growth of oyster mushrooms and classified their growth stages using YOLOv5, achieving an accuracy of 70% with a dataset of 4,271 annotated mushrooms. They also used Detectron2 to show the size of mushrooms on their substrate grow bags. Based on another dataset of 453 annotated mushrooms, they figured that mushrooms grow at an average rate of 5.22 days. But their images were taken in a small mushroom cultivation greenhouse, and they do not clarify where they obtained their data. Sujatanagarjuna *et al.* (2023) conducted a related study that created a scalable growing and harvesting system called MushR, which serves as a digital twin for gourmet mushroom production. They used Mask R-CNN in Detectron2 to find out how mature oyster mushrooms were, and they achieved an average accuracy 80% for large ones using a dataset of 3300 annotated mushrooms. They created a prototype for a mushroom harvesting system and made all of the models, images, and datasets available to the public. They carried out their work in a basic mushroom growing setting, using plastic buckets or pods for the harvesting system prototype and classifying maturity for each mushroom cap instead of for whole mushroom nodes. This limits the practicality and verifiability of their proposed system. Another research presents an innovative deep learning-based instance segmentation tracking pipeline in order to continuously monitor the growth of oyster mushroom nodes in a series of images taken at various times. The findings indicated that Mask R-CNN utilising ConvNeXt as a backbone attained superior performance in mushroom node instance segmentation (mAP=0.876, recall=0.858, F1-score=0.867) and tracking (MOTA=0.967). In this study, only images were used for monitoring and growth tracking, but environmental values should also be taken into account for mushroom growth and monitoring (Charisis *et al.*, 2025).

This study introduces a unique method for detecting the maturity of oyster mushrooms by utilising a combination of environmental data and images to train diverse machine learning algorithms, providing practical applications that surpass conventional camera-based systems. By combining sensor environmental data with mushroom images, the presented method in this paper becomes more flexible and useful for more than just growth analysis and maturity detection. This integrated method could also be used in robotic systems to accurately identify mushrooms, which would allow for damage-free harvesting and keep stable the quality of the mushrooms.

The main contributions of the study are:

• The development of a multisensory approach to determine oyster mushrooms maturity stage in greenhouse conditions by combining sensor environmental data with camera inputs. This makes the system more flexible and effective than traditional camera-based systems.

• Three distinct datasets of mushroom features were created and utilised for the classification of oyster mushroom maturity: one with manually extracted size features and two with automatically extracted size features using YOLOv8 and Detectron2 object detectors.

• Various classification algorithms, including Decision Trees (DT), Random Forest (RF), Multi-Layer Perceptron (MLP), Histogram-based Gradient Boosting Decision Trees (HistGBDT), and K-Nearest Neighbours (KNN), were trained, tested, and compared for maturity classification.

MATERIAL AND METHODS

This paper introduces a new multisensory approach for detecting the maturity of oyster mushrooms by utilising a combination of environmental data and images to train diverse machine learning algorithms. As shown in Figure 1, the methodology of the proposed system includes; building three new datasets which include the width, height, maturity classes of the mushroom nodes, as well as the temperature, humidity of the environments where the mushrooms are located, and the internal temperature of the mushroom bags at the same date and time. The difference between the datasets called manual and automatic; in the manual dataset, the height and width of the mushroom nodes are determined manually, and in the automatic datasets, these features are determined by the Detectron2 framework and YOLOv8 separately, by detecting the mushroom nodes automatically.

Data Processing and Preparation

Building upon a previously developed smart mushroom cultivation infrastructure (Elewi *et al.*, 2024), the present study focuses on multisensory maturity classification using combined visual and environmental features. Additionally, the environmental and visual data collected from multiple sensors and cameras were timestamped, compiled into a comprehensive dataset, and made publicly available (Duman *et al.* 2024). The dataset comprised 555 raw camera images from the greenhouse (see Figure 2). Getting only one image per day has the disadvantage that the growing days may not be counted precisely in some mushrooms (Moysiadis *et al.*, 2023). In this paper, this issue was addressed by capturing hourly images, which can introduce challenges in accurately tracking the growth of mushrooms.

For instance, in some cases, mushrooms may appear just a few hours after the image was taken, resulting in a shorter number of days of growth than the actual growth time. Similarly, mushroom mushrooms may have been harvested shortly before the image was taken, resulting in an underestimation of the days of growth.

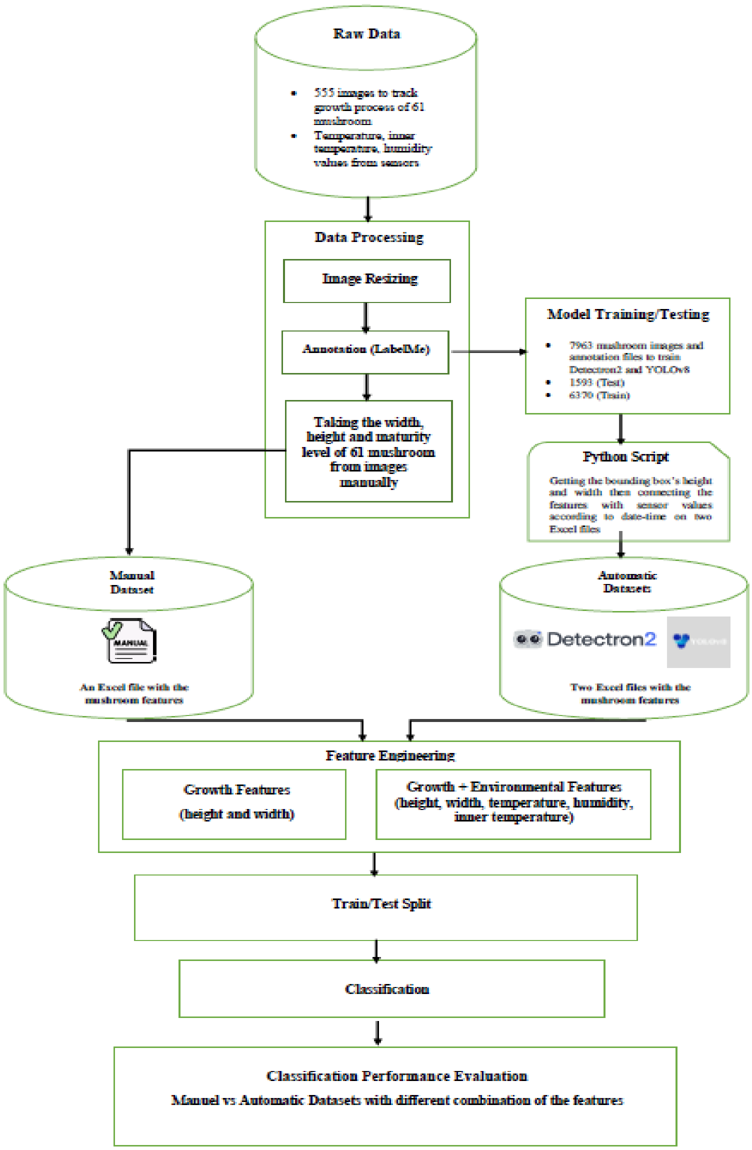

Figure 1. The Methodology of the Proposed Mushroom Maturity Classification System

Therefore, the importance was recognized of taking more frequent images of the substrate bags to yield improved results, offering a more precise assessment of mushroom growth rates and optimal harvesting times. After collecting the raw images, then 8,282 mushroom node images were extracted and annotated as “mature” and “immature” using LabelMe software (Russell *et al.*, 2007), offering their annotation files in three different (JSON, TXT and XML) formats to be used for training of YOLOv8 and Detectron2 separately in this study (Duman *et al.* 2024).

Figure 2. Sample Camera Raw Image from the Greenhouse

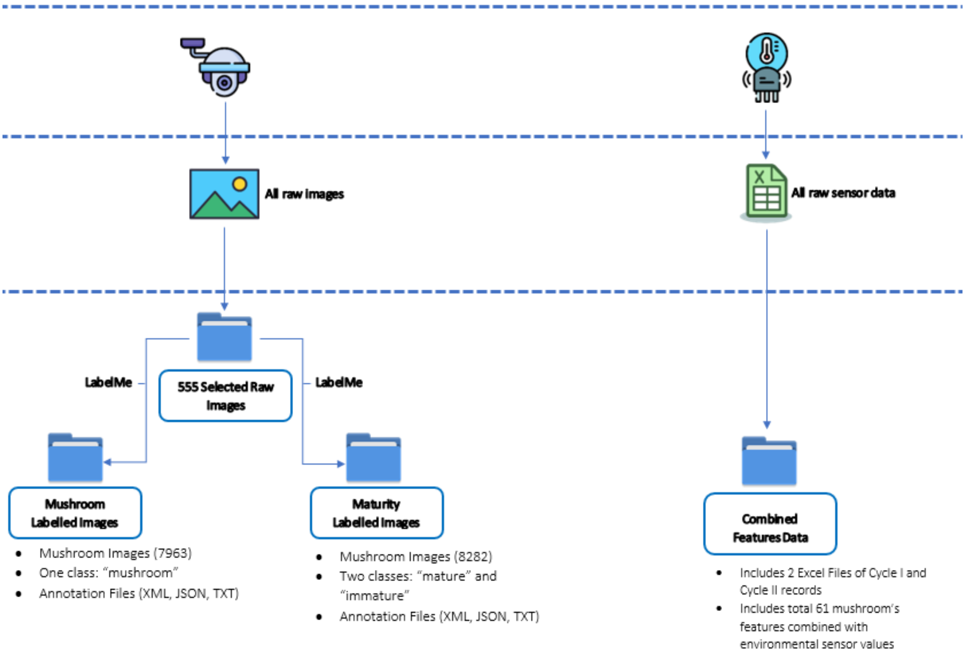

Figure 3. Data Collection Process

In sensor side, the data was collected from the smart mushroom farming system that designed and built with different types of cost-aware sensing nodes (Elewi *et al.,* 2024). Two cameras and a variety of sensors with different prices and functions were used to make it possible to monitor the farm remotely. The DHT21 sensor, which is relatively inexpensive and the SHT20 sensor, which is more reliable, were used to measure the temperature and humidity in the mushroom greenhouse. The CSS811 and SGP30 sensors were used to measure the air quality. The DS18B20 sensors were used to measure the temperature inside the mushroom composite bag. The data collection process of both images and sensor values can be shown as in Figure 3.

Manual Feature Extraction and Manual Dataset Building

In this phase, a manual dataset which includes size and environmental features of 61 mushrooms has built by using 1,703 images. Mushroom size features (width and height) were included manually from annotation files of these images, then a Python script connected these features to associated environmental features as humidity (hum), temperature (temp), and inner temperature (temp In) according to date/time. The main purpose here is to compare the maturity detection made using the manually extracted height and width features and the detection using automatically extracted height and width features via YOLOv8 and Detectron2 detectors together with the sensor data. This manual detection process serves as a valuable benchmark, allowing to compare and contrast the results obtained through the presented methodology with those produced by YOLOv8 and Detectron2. Figure 4 shows the manual extraction of a mushroom node from an image to get width and height of it.

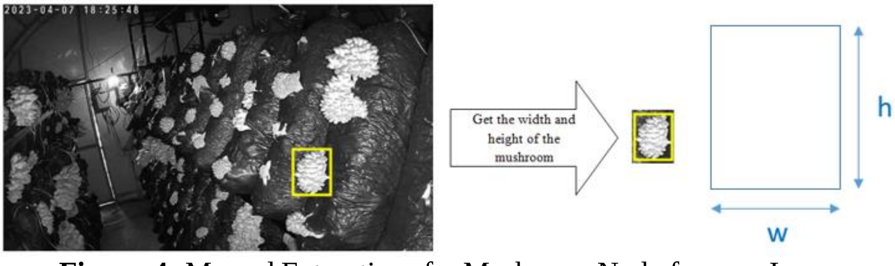

Figure 4. Manual Extraction of a Mushroom Node from an Image

The immature class encompassed mushrooms from their initial appearance up to one day before harvesting, while the mature class included mushrooms that were fully mature and ready for harvesting. In the dataset mature class represented as 1 and immature class represented as 0 corresponding to their respective growth stages. The maturity labels in this study were established based on harvesting time, representing practical cultivation choices typically employed by cultivators rather than direct physiological assessments. This operational definition was selected because harvest readiness represents the primary real-world criterion for mushroom maturity in commercial production.

Table 1 shows a sample of the dataset that was produced manually.

Table 1. Sample view of the manual dataset (the height and width features taken from bounding boxes which drawn manually)

mushroom id	width	height	date-time	maturity	temp	hum	temp in
1	38	47	2023-01-15 01:09:23	0	90.1057	9.76035	10.11
1	41	39	2023-01-15 08:32:36	0	77.0878	11.0136	12.0086
….	….	….	….	….	….	….	….
1	60	83	2023-01-17 04:23:02	1	92.2167	11.4935	12.2599
1	70	81	2023-01-17 08:28:19	1	90.5286	11.4191	11.9801

It is important to note that the dataset created manually was not used to train the model. Instead, it was only used as an independent benchmark dataset to compare the consistency of features extracted manually with those automatically extracted by the YOLOv8 and Detectron2 frameworks.

Automatic Dataset Building and Mushroom Detection with Detectron2

Detectron2 is an advanced platform which produced by Facebook AI Research (FAIR) for object detection and segmentation algorithms (Wu *et al.*, 2019). The Detectron2 framework's ability to handle different types of data and annotations, along with its impressive performance in object detection and segmentation tasks, makes it a valuable tool in the field of computer vision. It needs images and annotated databases that utilise the same format as the COCO dataset. Backbone is an important part of getting features from the input image with different architectures as ResNet, ResNeXt, and MobileNet. The Detectron2 framework is built upon convolutional neural networks, and the feature map of a single output for the convolution operation can be represented as Equation 1. Where the output feature map at position (i, j) for the k-th filter, x is input image, w is convolution filter, M and N are height and width of the filter.

$$y_{i,j,k} = \sum_{m=1}^{M} \sum_{n=1}^{N} x_i + m - 1, j + n - 1 \cdot w_{m,n,k} \tag{1}$$

In this study, after the dataset preparation and training pipeline configuration, the custom oyster mushroom dataset which includes 8,282 images were used to train the Detectron2 framework. The trained Detectron2 was tested on 1703 images to get essential features (mushroom height, width and maturity class from images) automatically. The Detectron2 framework detected mushrooms by drawing bounding boxes around them (see Figure 5). The features were extracted and added to dataset by a Python script that takes the width, height and maturity class of the mushroom node.

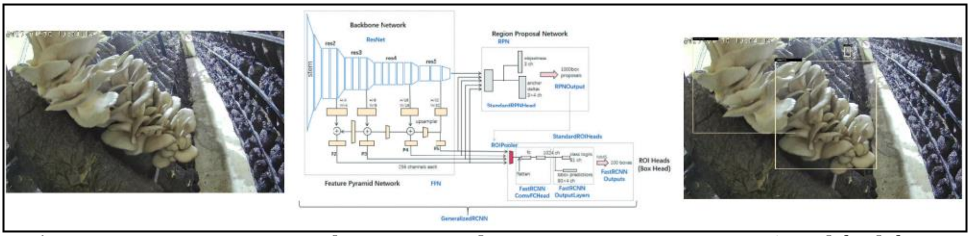

Figure 5. Detectron2 Architecture and Detection on an Image (Modified from Wu *et al.*, 2019)

The testing performance of the Detectron2 (model: Faster RCNN-50) on mushroom detection was evaluated by using some evaluation metrics. The Detectron2 framework provides a range of metrics to evaluate object detection performance. By considering precision at various recall levels and focusing on objects of different sizes, these metrics provide a comprehensive evaluation of the model's performance across different scenarios. These metrics (Zhang and Zhang, 2009); AP75, Average Precision at 75% overlap, which assesses precision at various recall levels by considering a detection as true positive (TP) if the Intersection over Union (IoU) overlap is at least 75%. APl, Average Precision (large), focusing on precision at different recall levels for objects with large sizes (area>962). APm, Average Precision (medium), measuring precision at different recall levels for objects with medium sizes (322<area<962). Aps, Average Precision (small), which calculates precision at different recall levels for objects with small sizes (area<322).

The Faster RCNN-50 that was used as Detectron2 model to detect mushroom nodes performs a successful detection with an AP75 score of 43.4, APl 51.2, APm 43, APs 21.1. These metrics show that the model performs best on large-sized mushroom nodes, with moderate success on medium ones, but struggles with small nodes. The height, width and maturity features values were added to the other dataset to compare results with the manual extraction. Table 2 shows the sample view of the dataset that was created automatically by getting width, height and maturity features from Detectron2.

Table 2. Sample view of the automatic dataset (the height, width and maturity features taken from bounding boxes which drawn by Detectron2)

mushroom id	width	height	date-time	maturity	temp	hum	temp in
18	213	206	2023-04-07 15:26:44	0	15.6588	99.11502	16.36695
18	233	226	2023-04-07 18:25:57	0	15.51356	99.37797	16.69209
….	….	….	……..	….	….	…..	…..
18	733	729	2023-04-11 15:15:20	1	17.092444	100	17.995555
18	748	734	2023-04-11 17:30:53	1	16.385185	100	17.818518

Automatic Dataset Building and Mushroom Detection with YOLOv8

In January 2023, Ultralytics introduced their latest YOLO model, YOLOv8, within the Ultralytics repository (Jocher *et al.*, 2023). In addition to pre-trained models, YOLOv8 enables users to create custom models adapted for their particular object detection requirements. In this process, users select and label the desirable object types in their training dataset, which is a component of data preparation. Training a custom model enhances accuracy and precision for object detection tasks specific to the user's application domain.

This work involved training YOLOv8 using a custom mushroom dataset. The dataset descriptor file's path was designated, outlining the dataset's location and format, and the "train" method was employed to train the YOLOv8 model. After training the YOLOv8 model, it was executed to classify mushrooms by drawing bounding boxes around them. Bounding boxes are crucial in this study as they mark the size of mushroom within the images. Figure 6 shows the YOLOv8 architecture by applying it on mushroom images.

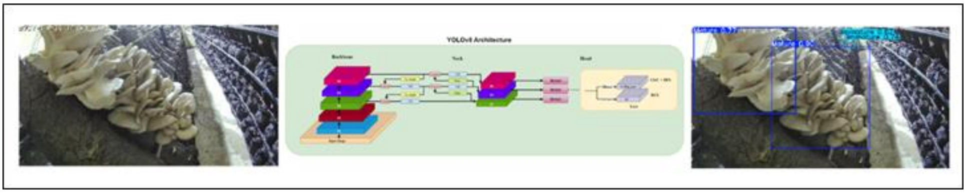

Figure 6. YOLOv8 Architecture and Detection on an Image (Modified from Jocher *et al.*, 2023)

Table 3. Sample view of the automatic dataset (the height, width and maturity features taken from bounding boxes which drawn by YOLOv8)

mushroom id	width	height	date-time	maturity	temp	hum	temp in
18	213	205	2023-04-07 15:26:44	0	15.6588	99.11502	16.36695
18	234	226	2023-04-07 18:25:57	0	15.51356	99.37797	16.69209
….	….	….	……..	….	….	…..	…..
18	748	734	2023-04-11 17:30:53	1	16.38518519	100	17.81851852
18	757	739	2023-04-11 21:15:54	1	15.56327434	100	17.2079646

YOLOv8 model provided accurate classification with 81.9% mean average precision (mAP) value at IoU 0.5, this indicates high detection accuracy when evaluated with a moderate overlap threshold. Also, the model achieved a strong detection performance, with a mAP@(0.5:0.95) of 56.7%, reflecting its consistent performance across stricter IoU thresholds. The results demonstrated that the model shows reliability not just at moderate overlap thresholds but also under harder evaluation settings.

The higher mAP@0.5 indicates better bounding box localization and object confidence, making YOLOv8 highly appropriate for real-time mushroom node detection where precision and speed are important. Table 3 presents a sample representation of the height, width, and maturity features extracted from bounding boxes created by YOLOv8.

Classification of Mushroom Maturity Class by Adding Environmental Features with Machine Learning Algorithms

In this study five different machine learning algorithms which include the Decision Tree (DT), Hist Gradient Boosting Classifier (HistGBDT), K-nearest Neighbor (KNN), Multilayer Perceptron (MLP), and Random Forest (RF) were used for classification. The Decision Tree (DT) is an algorithm which based on a tree-like data structure. The internal nodes in a Decision Tree show the dataset's attributes, the branches show the decision criteria, and the leaf nodes show the final outcomes. This tree structure can efficiently encode Boolean values, covering both true and false states (Fratello and Tagliaferri, 2019). K Nearest Neighbour (KNN) is a machine learning algorithm works by comparing the new data to the old data to see how similar they are. It puts the new data into the category that is most similar to the ones that are already there (Sun and Huang, 2010). The Hist Gradient Boosting Classifier (HistGBDT) is a version of the standard gradient boosting decision tree (GBDT) method, designed to accelerate the learning process and enhance the model's predictive performance (Tamim Kashifi and Ahmad, 2022). It comes from applying gradient boosting to decision trees and using a better base learner called the piecewise linear tree, which uses linear functions to make predictions at the leaves. This method improves the weak classifier's loss function by using a learning rate, which decreases the loss for the next classifier. The Multilayer Perceptron (MLP) classifier is an important part of neural networks because it has a multi-layered structure. The hidden layers of a neural network give it the ability to learn complex patterns and representations in data. This allows it capture detailed features and relationships in the incoming data (Rumelhart *et al.*, 1986). Random Forest (RF) is a machine learning technique used for classification and regression. Because it uses more than one decision tree model, it is often called an ensemble classifier. This classifier combines several decision trees made from different parts of the dataset and averages their results to make predictions more accurate (Breiman, 2001).

As mentioned in the previous sections, the three different datasets were created as one manual and two automatic datasets. These datasets were used to classify the mushroom maturity class by the height, width, humidity, temperature and inner temperature features. Model evaluation was performed using a consistent randomized train–test split while preserving class balance across maturity categories. The same data partitioning strategy was applied to all classifiers and datasets to ensure fair comparison. This procedure was repeated 200 times using different random seeds to reduce sampling bias and improve robustness of performance estimation. Classification metrics were computed for each iteration, and the final reported results correspond to the mean performance values across all repetitions. The algorithms that were used to make classification were evaluated with precision, recall, F-1 Score, and accuracy metrics which are key metrics for evaluating classification models. Precision as shown in Equation 2 is the ratio of true positives(TP) to true positives plus false positives(FP), showing how few false positives the model makes (Ferrer, 2022). Recall (sensitivity) is the ratio of true positives to true positives plus false negatives(FN), reflecting how well the model identifies actual positives. Equation 3 represents the calculation formula of the recall (Ferrer, 2022). Accuracy as shown in Equation 4 is the ratio of true positives plus true negatives(TN) to total instances, indicating overall prediction correctness (Ferrer, 2022). F-1 Score which represented in Equation 5 is the harmonic mean of precision and recall, providing a balanced measure between them (Ferrer, 2022).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$F1\ Score = 2 \text{ x } \frac{Precision\ x\ Recall}{Precision+Recall} \quad (5)$$

The classification algorithms were applied on the datasets separately and evaluated with the mentioned evaluation metrics. The results will be discussed in the following section.

RESULTS AND DISCUSSION

The main purpose of the study is to classify the maturity class of mushroom nodes by using these three datasets separately. After the classification, the results between manual and automatic datasets which includes the features detected by Detectron2 and YOLOv8 were compared. During the classification, environment values were included in mushroom node features with different combinations. One of the classification algorithms used is the HistGBT classifier. With this classifier, maturity classes were first tried to be classified by using only the height and width of the mushroom nodes. Afterwards, temperature and humidity values, which are the most important environmental values, were included among the features and reclassified. As seen in Table 4 and 5 the HistGBT algorithm, which works with an accuracy rate of 81% on the manual dataset, increases the success rate to 92.8% when these two features are added. Then, the inner temperature value of the bags was added as the last feature. As seen in Table 6, this feature reduced the classification success of the algorithm. The reduction in classification performance following the inclusion of inner bag temperature as a feature can be explained through various supplementary hypotheses. First, measuring the temperature inside the bags could add noise to the sensors because the micro-environment inside the bags changes a lot and the sensors are sensitive to where they are placed. Unlike sensors that measure the environment around them, sensors that measure the inside of an object may give very different readings even at the same growth stage. This makes it harder for machine learning models to get consistent signals. Second, the inner temperature information may not be very useful because it is already included in the feature set along with the temperature and humidity of the surrounding area.

Table 4. Evaluation results of the mushroom maturity classification of the Manual and Automatic Datasets (Note. Using only width and height features of mushroom)

Classifier	**Accuracy**	**Precision**	**Recall**	**F1 Score**
HistGBDT (Manual)	***0.84***	***0.84***	***0.84***	***0.84***
HistGBDT (Detectron2)	0.83	0.82	0.82	0.82
HistGBDT (YOLOv8)	0.83	0.82	0.84	0.83
RF (Manual)	0.83	0.83	0.83	0.83
RF (Detectron2)	0.81	0.80	0.80	0.80
RF (YOLOv8)	0.82	0.83	0.82	0.82
DT (Manual)	0.82	0.83	0.82	0.82
DT (Detectron2)	0.80	0.81	0.81	0.82
DT (YOLOv8)	0.82	0.82	0.82	0.82
KNN (Manual)	0.81	0.79	0.81	0.80
KNN (Detectron2)	0.80	0.79	0.80	0.79
KNN (YOLOv8)	0.81	0.79	0.81	0.80
MLP (Manual)	0.83	0.82	0.83	0.83
MLP (Detectron2)	0.81	0.81	0.80	0.80
MLP (YOLOv8)	0.82	0.82	0.83	0.82

Random Forest classifier was used as another classification algorithm. As seen in Table 4, the random forest classifier's success in classifying mushroom maturity using only the width and height of mushroom nodes has an accuracy rate of 83% in the manual dataset and 81% in the Detectron2 automatic dataset and 82% in the YOLOv8 automatic dataset. Afterwards, reclassification was made with adding temperature and humidity features from the environmental values which increase the success of the algorithm (see Table 5). Then, the inner temperature feature was added to these features, the classification success decreased to a certain extent (see Table 6). Decision tree classifier, which is one of the tree classifier models, was used as another classification algorithm.

Similarly, the success increased when temperature and humidity features were added to the environmental values (see Table 5). However, the inner temperature value again reduced the model success (see Table 6).

Table 5. Evaluation results of the mushroom maturity classification of the Manual and Automatic Datasets (Note. Using width, height, temperature and humidity features of mushroom)

Classifier	Accuracy	Precision	Recall	F1 Score
HistGBDT (Manual)	***0.92***	***0.91***	***0.91***	***0.91***
HistGBDT (Detectron2)	0.91	0.90	0.90	0.90
HistGBDT (YOLOv8)	0.91	***0.91***	***0.91***	***0.91***
RF (Manual)	0.88	0.87	0.87	0.87
RF (Detectron2)	0.85	0.84	0.85	0.85
RF (YOLOv8)	0.87	0.86	0.86	0.86
DT (Manual)	0.83	0.83	0.83	0.83
DT (Detectron2)	0.82	0.82	0.82	0.82
DT (YOLOv8)	0.83	0.82	0.82	0.82
KNN (Manual)	0.84	0.83	0.83	0.83
KNN (Detectron2)	0.82	0.82	0.82	0.82
KNN (YOLOv8)	0.83	0.82	0.82	0.82
MLP (Manual)	0.85	0.83	0.85	0.84
MLP (Detectron2)	0.83	0.82	0.83	0.82
MLP (YOLOv8)	0.84	0.83	0.85	0.84

Unlike other algorithms, KNN gave more successful results when all features were used. A possible reason of this that KNN relies heavily on the similarity of instances. KNN may perform well in classification when the features exhibit significant relevance in proximity or similarity. Alternative classifiers such as Random Forest, Decision Trees, and Gradient Boosting may not properly detect this resemblance, particularly when the interactions between features and the target variable are complex. Finally, MLP classifier, an artificial neural network-based algorithm, was used as a classifier. The success of the MLP classifier is given in the Table 4, 5, and 6. Unlike KNN, MLP also produced a similar result to the other three classifiers.

Table 6. Evaluation results of the mushroom maturity classification of the Manual and Automatic Datasets (Using width, height, temperature, inner temperature and humidity features of mushroom)

Classifier	Accuracy	Precision	Recall	F1 Score
HistGBDT (Manual)	***0.88***	***0.87***	***0.88***	***0.87***
HistGBDT (Detectron2)	0.87	0.86	0.86	0.86
HistGBDT (YOLOv8)	0.87	***0.87***	0.86	0.86
RF (Manual)	0.87	0.86	0.87	0.86
RF (Detectron2)	0.85	0.83	0.83	0.83
RF (YOLOv8)	0.87	0.86	0.86	0.86
DT (Manual)	0.82	0.82	0.81	0.82
DT (Detectron2)	0.79	0.80	0.79	0.79
DT (YOLOv8)	0.82	0.82	0.81	0.82
KNN (Manual)	0.86	0.85	0.85	0.85
KNN (Detectron2)	0.84	0.84	0.83	0.84
KNN (YOLOv8)	0.85	0.85	0.84	0.85
MLP (Manual)	0.84	0.84	0.83	0.84
MLP (Detectron2)	0.82	0.82	0.81	0.82
MLP (YOLOv8)	0.82	0.82	0.82	0.82

According to the results, the HistGBDT classifier showed higher success than the other four classifiers. This can be explained as it adeptly captures complex non-linear relationships between features and the target variable through its sequential ensemble of decision trees, correcting errors iteratively. This capability enables it to model intricate decision boundaries better than RF or DT. When the results obtained are examined, it can be said that the environmental temperature and humidity, as well as the height and width features of the mushroom node, are important in classifying the maturity class of a mushroom node.

CONCLUSIONS

This study introduced a comprehensive approach that aims to automate the classification of mushroom nodes using different features. By creating three different datasets containing features extracted by human and detected by Detectron2, YOLOv8 object recognition frameworks, it was examined how useful these datasets were for mushroom maturity classification. The manual dataset consisted of 61 mushroom instances and was intentionally designed as an evaluation benchmark rather than a training dataset. Therefore, its limited size does not affect model learning capacity but may constrain the statistical representativeness of human–automatic feature comparisons. In addition, it was revealed how the classification success changes when environmental temperature, humidity and inner temperature values are added as features which greatly affect mushroom growth processes. Compared with all classifiers, the overall classification success of the HistGBDT classifier was high on manual and automatic datasets. Also, the results show that especially temperature and humidity values significantly increased the classification success. This indicates that environmental factors also have a high impact on maturity classification. In addition to this, it was observed that the classification results using the automatic datasets are very close to the classification made with the manual dataset. This shows that YOLOv8 and Detectron2 are successful in object detection and bounding box drawing.

In conclusion, this proposed multisensory approach for detecting mushroom maturity class in greenhouses, offering practical applications that transcend conventional camera-based systems. By integrating sensor data alongside cameras, the approach enhances its adaptability and efficiency. Future work may incorporate physiological maturity indicators to evaluate cross-greenhouse transferability and reduce potential temporal bias. A notable application lies in yield prediction, where the relation between cameras and sensor data enables close monitoring of greenhouse conditions, enabling yield forecasts days prior to harvest. Moreover, this integrated methodology can be integrated into robotics systems to accurately locate mushroom nodes, thereby facilitating damage-free harvesting and ensuring the preservation of mushroom quality in the future.

REFERENCES

Adebayo, S., Aworinde, H.O., Olufemi, O.O., Osueke, C.O., Adeniyi, A.E. & Aroba, O.J. (2025). Understanding mushroom farm environment using TinyML-based monitoring devices. Environ. Res. Commun., 7(4): 045014. https://doi.org/10.1088/2515-7620/adc5cd

Badoni, P. & Siddiqui, S.A. (2025). Metamorphosis of mushroom production from tradition to automation. Discover Appl. Sci., 7: 974. https://doi.org/10.1007/s42452-025-07517-w

Ben Mansour, A., Ababsa, N. & Berkani, C. (2025). Waste-based substrates for edible mushroom production: A sustainable approach to food and waste management. Euro-Mediterr. J. Environ. Integr., 10: 2473–2485. https://doi.org/10.1007/s41207-025-00818-x

Breiman, L. (2001). Random forests. Mach. Learn., 45: 5–32. https://doi.org/10.1023/A:1010933404324

Charisis, C., Nuwayhid, S. & Argyropoulos, D. (2025). A novel Mask R-CNN-based tracking pipeline for oyster mushroom cluster growth monitoring in time-lapse image datasets. Comput. Electron. Agric., 237: 110590. https://doi.org/10.1016/j.compag.2025.110590

Chen, L., Qian, L., Zhang, X., Li, J.Z., Zhang, Z.J. & Chen, X.M. (2022). Research progress on indoor environment of mushroom factory. Int. J. Agric. Biol. Eng., 15(1): 25–32. https://doi.org/10.25165/j.ijabe.20221501.6741

Elewi, A., Hajhamed, A., Khankan, R., Duman, S., Souag, A. & Ahmed, A. (2024). Design and implementation of a cost-aware and smart oyster mushroom cultivation system. Smart Agric. Technol., 8: 100439. https://doi.org/10.1016/j.atech.2024.100439

Ferrer, L. (2022). Analysis and comparison of classification metrics. arXiv preprint, arXiv:2209.05355. https://doi.org/10.48550/arXiv.2209.05355

Fratello, M. & Tagliaferri, R. (2019). Decision trees and random forests. In Encyclopedia of Bioinformatics and Computational Biology, pp. 374–383. Amsterdam, Elsevier. https://doi.org/10.1016/B978-0-12-809633-8.20337-3

Jayaraman, S., Yadav, B., Dalal, R.C., Naorem, A. & Sinha, N.K. (2024). Mushroom farming: A review focusing on soil health, nutritional security and environmental sustainability. Farm Syst., 2(3): 100098. https://doi.org/10.1016/j.farsys.2024.100098

Jocher, G., Chaurasia, A. & Qiu, J. (2023). Ultralytics YOLOv8 (Version 8.0.0) (computer software). Available at https://github.com/ultralytics/ultralytics

Lu, C.P. & Liaw, J.J. (2020). A novel image measurement algorithm for common mushroom caps based on convolutional neural network. Comput. Electron. Agric., 171: 105336. https://doi.org/10.1016/j.compag.2020.105336

Lu, C.P., Liaw, J.J., Wu, T.C. & Hung, T.F. (2019). Development of a mushroom growth measurement system applying deep learning for image recognition. Agronomy, 9(1): 32. https://doi.org/10.3390/agronomy9010032

Moysiadis, V., Kokkonis, G., Bibi, S., Moscholios, I., Maropoulos, N. & Sarigiannidis, P. (2023). Monitoring mushroom growth with machine learning. Agriculture, 13(1): 223. https://doi.org/10.3390/agriculture13010223

Paudel, N. & Bhatta, J. (2022). Mushroom classification using Random Forest and REP Tree classifiers. Nepal J. Math. Sci., 3(1): 111–116. https://doi.org/10.3126/njmathsci.v3i1.44130

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. (2022). IoT enabled mushroom farm automation with machine learning to classify toxic mushrooms in Bangladesh. J. Agric. Food Res., 7: 100267. https://doi.org/10.1016/j.jafr.2021.100267

Raut, J.K. (2019). Current status, challenges and prospects of mushroom industry in Nepal. Int. J. Agric. Econ., 4(4): 154. https://doi.org/10.11648/j.ijae.20190404.13

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986). Learning representations by back-propagating errors. Nature, 323: 533–536. https://doi.org/10.1038/323533a0

Russell, B.C., Torralba, A., Murphy, K.P. & Freeman, W.T. (2007). LabelMe: A database and web-based tool for image annotation. Int. J. Comput. Vis., 77: 157–173. https://doi.org/10.1007/s11263-007-0090-8

Sangeeta, Sharma, D., Ramniwas, S., Mugabi, R., Uddin, J. & Nayik, G.A. (2024). Revolutionizing mushroom processing: Innovative techniques and technologies. Food Chem. X, 23: 101774. https://doi.org/10.1016/j.fochx.2024.101774

Singh, H.D., Patel, T., Singh, H.J. & Pal, A. (2025). Oyster mushroom farming with IoT and artificial intelligence: A comprehensive review. J. Agric. Eng. (India), 62(2): 293–306. https://doi.org/10.52151/jae2025622.1920

Sujatanagarjuna, A., Kia, S., Briechle, D.F. & Leiding, B. (2023). MUSHR: A smart, automated, and scalable indoor harvesting system for gourmet mushrooms. Agriculture, 13(8): 1533. https://doi.org/10.3390/agriculture13081533

Sun, S. & Huang, R. (2010). An adaptive K-nearest neighbor algorithm. In Fuzzy Systems and Knowledge Discovery, pp. 91–94. https://doi.org/10.1109/FSKD.2010.5569740

Tamim Kashifi, M. & Ahmad, I. (2022). Efficient histogram-based gradient boosting approach for accident severity prediction with multisource data. Transp. Res. Rec., 2676(6): 236–258. https://doi.org/10.1177/03611981221074370

Tongcham, P., Supa, P., Pornwongthong, P. & Prasitmeeboon, P. (2020). Mushroom spawn quality classification with machine learning. Comput. Electron. Agric., 179: 105865. https://doi.org/10.1016/j.compag.2020.105865

Vinci, G., Prencipe, S.A., Pucinischi, L. & Perrotta, F. (2023). Sustainability assessment of waste and wastewater recovery for edible mushroom production through an integrated nexus. Sci. Total Environ., 903: 166044. https://doi.org/10.1016/j.scitotenv.2023.166044

Wang, Y., Yang, L., Chen, H., Hussain, A., Ma, C. & Al-Gabri, M. (2022). Mushroom-YOLO: A deep learning algorithm for mushroom growth recognition based on improved YOLOv5. Proc. IEEE Int. Conf. Ind. Inform., pp. 239–244. https://doi.org/10.1109/INDIN51773.2022.9976155

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y. & Girshick, R. (2019). Detectron2 (computer software). Available at https://github.com/facebookresearch/detectron2

Zhang, E. & Zhang, Y. (2009). Average precision. In Encyclopedia of Database Systems, pp. 192–193. New York, Springer. https://doi.org/10.1007/978-0-387-39940-9_482

Zhu, F., Sun, Y., Zhang, Y., Zhang, W. & Qi, J. (2023). An improved MobileNetV3 mushroom quality classification model using images with complex backgrounds. Agronomy, 13(12): 2953. https://doi.org/10.3390/agronomy13122924