ICIC Express Letters
Part B: Applications
Volume **13**, Number **12**, December **2022**

ICIC International ©2022 ISSN 2185-2766
pp. 1301–1312

FUNGAL DISEASE DETECTION SYSTEM FOR FAIRY MUSHROOMS USING DEEP LEARNING, ROBOTICS AND IOT FOR REAL SMART FARMING

Patcharee Sriputta and Suchart Khummanee*

Department of Computer Science
Faculty of Informatics
Mahasarakham University
Khamriang Sub-District, Kantarawichai District, Maha Sarakham 44150, Thailand
sriputta29@gmail.com; *Corresponding author: suchart.k@msu.ac.th

Received April 2022; accepted June 2022

Abstract. *Mushrooms are highly nutritious and vital cash crops in many countries. However, a critical problem for mushroom cultivation is fungal diseases. They can inflict widespread damage on mushrooms within a relatively short period (within 48 hours), even though mushrooms are grown in precision farming systems. Besides, mushroom spores are also harmful to the health of mushroom farmers. In this paper, a fungal disease diagnosis and detection system (Case study: Fairy mushroom) is developed using various deep learning techniques, working with robots (Photographic robots) and precise Internet of Things (IoT) systems under the conditions of an intelligent farming environment. The aims are to reduce the cost of mushroom wastage caused by fungal diseases, alleviate the health problems of farmers caused by mushroom spores, and address the vulnerabilities of the current intelligent farm system. The system consists of seven parts: 1) collecting fungal disease data from real farms, 2) designing and developing intelligent farms and photography robots, 3) data pre-processing, 4) image pre-processing, 5-6) choosing a CNN model, and 7) real-time prediction. The classification algorithms to recognize fungal diseases include DenseNet201, ResNet50, InceptionV3, and VGG19 based on the image database of 1,000 images of non-fungus and fungal mushrooms equally. The experimental results of the classification of fungal diseases show that DenseNet201 has the highest accuracy of 89.74% (DenseNet201 is most suitable for detecting fungal diseases for existing smart farms). In particular, the proposed system can detect fungal diseases rapidly in only 1-6 hours. As a result, mushroom growers can reduce the number of mushrooms damaged by fungal diseases and the risk of direct exposure to mushroom spores.*
Keywords: Mushroom, Fairy mushroom, Fungal disease, Deep learning, Smart farming, Robots, IoT

1. **Introduction.** Nowadays, mushrooms are becoming increasingly popular consumption. They have high nutritional value in carbohydrates, proteins, fats, minerals, and vitamins. Therefore, they are suitable for patients with a high risk of death, such as liver, lung abscess heart disease, and hypertension. Unfortunately, naturally occurring mushrooms cannot produce all year round (high yield during the rainy season). Thus, they are insufficient to consume. Growing mushrooms all year round can solve this problem through a closed ecosystem called smart farming. It can precisely control environmental conditions such as temperature, humidity, lighting, and carbon dioxide. However, we found that the intelligent farms can precisely control their environment, the fungal disease can still occur. For example, mushroom fungal disease in smart farms is caused by moisture accumulated during fogging to reduce temperature and increase the humidity inside the greenhouse. Fungal diseases are hazardous to mushrooms as they cause to stop mushroom growing. Moreover, the fungus spread inside the greenhouse is very rapid (within

DOI: 10.24507/icicelb.13.12.1301

48 hours, the whole mushrooms may be infected). Thus, the farmers must immediately remove all those infected mushrooms from the greenhouse; otherwise, all mushrooms will lose. In addition, the mushroom spore is very dangerous to farmers' activities such as picking and growing surreys of mushrooms as it causes several diseases such as lung abscess, lung disease, and allergies. Besides, their average time spent inside the greenhouse is about 1 to 3 hours per day. Therefore, farmers inhale large quantities of spores, harming long-term health.

Fungi and spores of mushrooms have severe effects on the health and cost of cultivation for farmers. Cultivating mushrooms in innovative farm ecosystems has dramatically reduced the incidence of fungi. However, there is still a potential for fungal disease during the temperature rise and fall of the greenhouse. Besides, farmers are still in direct contact with fungi and spores of the mushrooms. Therefore, this paper contributes a prototype to detecting fungal diseases in only six hours to reduce the aforementioned problems and alert farmers when detected in intelligent farms.

This paper is organized as follows: Section 2 reviews related literature, Section 3 designs and develops our prototype system, Section 4 presents the results of the experiment and evaluation, and the last one is the summary of the paper (Section 5).

2. **Background and Related Work.** In this section, we discuss related research to analyze various fungal diseases in plants such as banana leaves, corn leaves, tomatoes, peppers, and mushrooms, in order to study fungal disease patterns, analysis techniques, and tools. Over the years, deep learning [1-3] (CNN) has been increasingly applied in agricultural applications in planting, plant growth analysis, and plant disease analysis because it can predict real-life situations with high accuracy. As a result, farmers can effectively monitor, treat and prevent disease as follows.

Mohanty et al. [4] addressed the development guidelines' accuracy in image classification and analysis of diseased plants. They used image data from Plant Village stored at the GitHub Repository. Fifty-four thousand three hundred plant leaf images were divided into 38 classes of diseased and non-pathogenic images. They used CNN techniques (AlexNet and googLeNet) for classification and analysis. The results showed that CNN could analyze plant diseases with an accuracy of 99.35%. Amara et al. [5] proposed a convolution neural network (CNN) called LeNet architecture deep learning approach to automatically identify plant diseases in banana leaves using image datasets from the PlantVillage project [4,6]. The result showed that the experiment with color images had a maximum accuracy of 99.72%. Ferentinos [3] developed a neural network model for detecting plant diseases using an open database of Hughes and Salathé [6]. The data consists of 87,848 images from 25 plant species divided into 588 classes. For classification and analysis, they used CNN techniques (AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat, and VGG). The results showed that VGG had the highest accuracy of 99.53%. In addition, many researchers have used artificial intelligence to analyze crop diseases in innovative farm systems.

Kitpo et al. [7] presented an IoT system with bots to alert the growing stage of tomatoes. They used images data set from Shinchi Agri Green, the greenhouse in Fukushima, Japan. This image data set from real-images was captured in greenhouse amount 25-day which selected image amount 263 images, and images were divided into six stages because stages are discoloration of tomatoes indicated the growth. They used Faster R-CNN techniques for object detection, K-Means clustering for segmentation, and SVM classifier to classify tomato growing stages. The results showed that the accuracy of the tomato growth alert was 91.5%. However, this research should use augmentation techniques to increase the dataset. Chen et al. [2] proposed automated plant disease recognition. The plants used in the experiment were rice and corn. They used image rice leaf diseases of 500 images, image corn leaf diseases of 466 images, and image plant leaf of 1,000 images. They divided the image data for training and testing as 70% : 30%, using four classical models (DenseNet-201, ResNet-50, the InceptionV3, and VGG-19). The results showed that INC-VGGN gave an accuracy of 92.00%. This work contains a variety of image data, and most importantly, the experimental images are all images taken from actual conditions with various models. Khummanee et al. [8] built an intelligent farm to control the orchid growth environment and orchid disease using fuzzy logic. The results showed that the orchids continued to grow well throughout the year (average growth rate of approximately 27.38 cm per week) and were less disease-free. Moreover, Wiangsamut et al. [9] developed a system of conversation with orchids to inquire about the health of orchids. The proposed chat system gives an average accuracy of interacting with the owner of the orchid grower at 71 percent. Based on a comprehensive review of the above research, we design and implement a prototype to detect and notify fungal diseases (Case study: Fairy mushroom), detailed in the next section.

3. **Designing and Developing Our Prototype System.** This section has designed a framework for detecting and alerting when the disease occurs in farms consisting of seven steps, as shown in Figure 1. Each step is illustrated as follows.

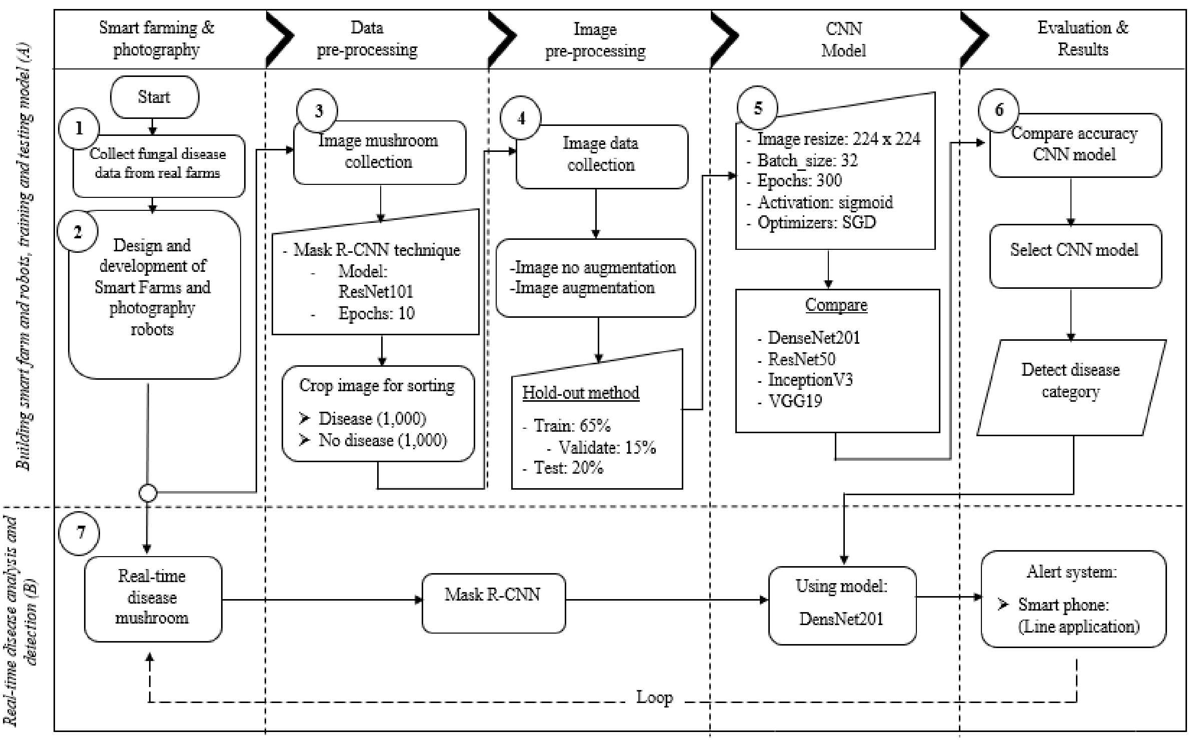
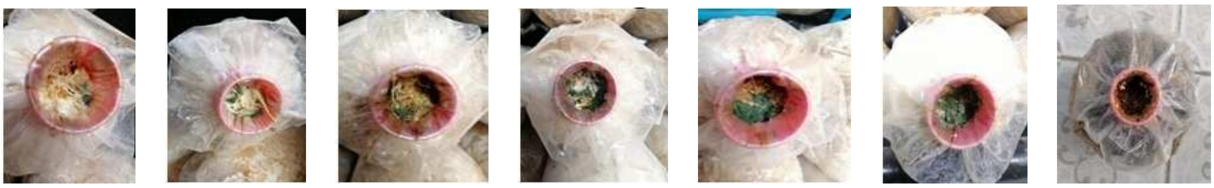


Figure 1. A proposed framework for mushroom detection and alarm for a smart farm

1) Collect fungal disease data from real farms. In this research, we visited several areas of natural mushroom farming to study and collect the information and problem on the incidence of mushroom fungal disease. The study results revealed the two leading fungal disease causes: contamination of the plant material during the packing process and accumulated moisture inside the greenhouse during cultivation. According to Figure 2, it shows the fungal disease patterns from 1 to 48 hours:

(a) shows the beginning of the fungal disease in the range of 1-6 hours,
(b) begins to spread to nearby areas over approximately 6-12 hours,
(c) spreads almost the entire area (approximately 12-18 hours),
(d) spreads throughout the planting area (about 18-24 hours),
(e) begins to spread to other areas within the planting bag (about 24-30 hours),

(f) spreads in about one-third of the planting bag (around 30-36 hours),
(g) finally, it eventually spreads throughout the planting bag (over 36 hours).
After that, it rapidly spreads to other planting bags within the farm; if the farmer does not detect it within 48 hours, it can damage all mushrooms.



(a) 1-6 hrs (b) 6-12 hrs (c) 12-18 hrs (d) 18-24 hrs (e) 24-30 hrs (f) 30-36 hrs (g) 36-48 hrs

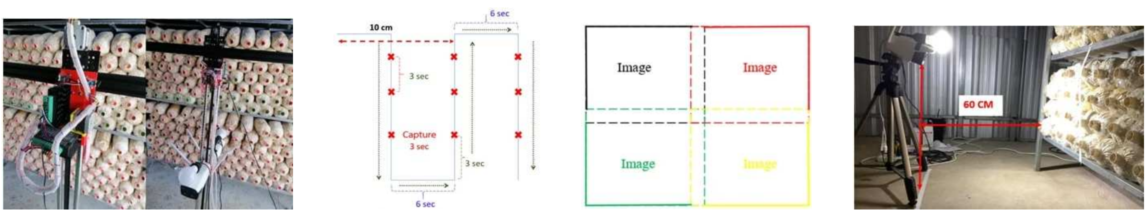
FIGURE 2. Examples of the severity of green mold disease spreading

2) Design and development of smart farms and photography robots. An overview of a real innovative farm for mushroom cultivation is shown in Figure 3. The equipment in the innovative farm includes ventilation fans, a water pump, a cooling panel (EVAP), a solar panel, a solar control cabinet (Inverter), a 220-volt electrical system, IP cameras, a lighting system as illustrated in Figure 3(a). IoT equipment is microcontrollers, temperature, humidity sensors, soil moisture, fogging, and lighting (Figure 3(b)). Temperature and humidity inside the smart farms are 28-32 Celsius (°C) and 60%-80%, respectively. The room for controlling the system contains the electrical system, sensor receivers, and the computer system for AI processing, as shown in Figures 3(c) and 3(d).

(a) (b) (c) (d)

FIGURE 3. Innovative farm structure for mushroom cultivation

This research proposes to detect fungi as fast as possible where farmers are less exposed to mushroom spores by using robots to take pictures inside the intelligent farming, namely IBOT. The captured mushroom images are then processed with deep learning (CNN [4,10-13]) to predict fungal diseases over the wireless network. The IBOT is illustrated in Figure 4(a). The robot movement pattern is a square wave, as shown in Figure 4(b). To protect against the distortion and blurring problem, we need to shoot the overlapping images to solve the edge blurring problem, as shown in Figure 4(c) (Overlap 15%: 108×230 pixel). The picture's width is about 17.06 cm, and the length is 33.87 cm. We calculate the optimal distance and brightness for shooting with an IP camera (Full HD 1080) by



(a) (b) (c) (d)

FIGURE 4. Overview of the photography robot (IBOT) and its parameterization

experimenting with four shooting distances: 50, 60, 70, and 80 centimeters (cm). As a result, from the experiment, the best distance from the camera to the mushroom that the most precise picture is 60 cm, and the light is 12 watts, as shown in Figure 4(d).

3) Data pre-processing. This section describes how to crop the mushroom images using Mask R-CNN [14,15] (ResNet101). Figure 5(a) is an example of a mushroom picture captured by the IBOT inside the innovative farm with dimensions of 720 × 1,280 pixels. In fact, mushroom fungus always occurs in front of the mushroom lump (Circle area) and then gradually spreads to other areas. Therefore, this area is chosen as a target of the Mask R-CNN [12] for cropping. As a result, the output image cropped by Mask R-CNN is permanently fixed at 200 × 200 pixels, and the number of cropped images ranges from 13 images with the following steps.

(a) An original mushroom image from IBOT

(b) An image annotated with polygon shapes

Figure 5. Examples of original mushroom images and annotated images

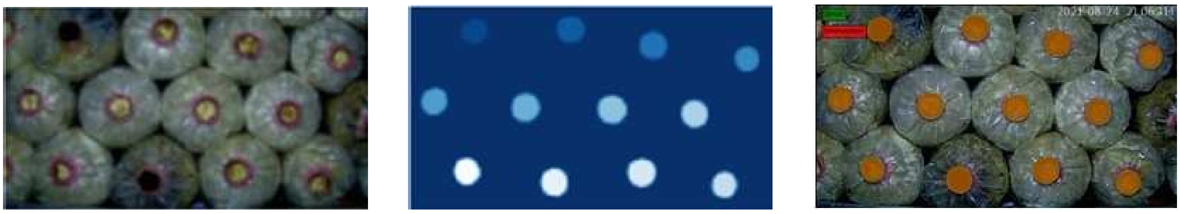
Annotation process: This step marks the circles in the original mushroom image with polygons before cropping. The total number of images used for marking is 250 images divided into two groups: 200 images for training (80%) and 50 images for testing (20%). All images are annotated in polygon shapes at the center of each mushroom image as shown in Figure 5(b). The result of this step is the positions of the polygon images exported in JSON format.

Cropping process by Mask R-CNN: The exported JSON file has been imported into this process of the training model. We set the number of trainings to 10 epochs, and all other values are based on the default Mask R-CNN (in Table 1). It starts by using a selective search mechanism to extract the region of interest (ROI), where each ROI [15] is a rectangle that may represent the region of the object in the image. The number of epochs is determined by only ten epochs due to the mean data consistency or Intersection over Union (IoU) obtained from the experiments 10, 20, and 30 epochs that are not different. The objects of the mushroom image after Mask R-CNN (using ResNet101) processing successfully are shown in circles, as shown in Figure 6(c). After this process ends, the trained model is ready for testing. The results showed that the Mask R-CNN could find all circular objects (Center of the mushroom) and was consistent with IoU [16-19], in which the mean consistency of the data set was at 0.86 (86%). The IoU value [20] greater than or equal to 50% (0.5) is acceptable (Positive overlap threshold), and else is unacceptable (Negative overlap threshold).

We import both the test image and the annotated file to evaluate the model performance in the model testing stage. The evaluation results are shown in Figure 6. The results of the model test can be divided into three parts: (a) the result of the annotation mask, (b) the result of the Mask R-CNN technique (ResNet101), and (c) the ability to find objects using the Mask R-CNN technique. The results showed that the Mask R-CNN could find all circular objects (Center of the mushroom). Besides, the IoU value is 0.86 (86%), which is also greater than the default (IoU default value = 0.5). After locating the desired objects with the Mask R-CNN technique, the next step is to crop the mushroom lump pages one by one image. The method of cropping the image is as follows.

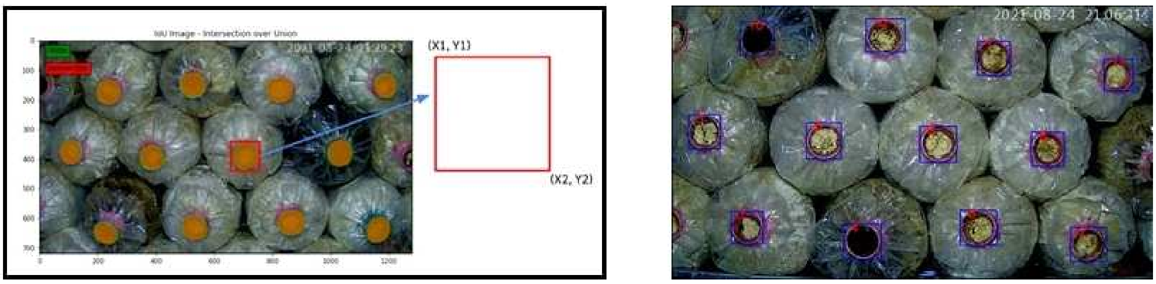
TABLE 1. Setting parameters Mask R-CNN

No.	Parameter	Value	Remark
1	BACKBONE	resnet101	Model
2	LEARNING_RATE	0.001	Learning rate
3	IMAGE_SHAPE	[512 512 3]	Shape image
4	MASK_SHAPE	[28, 28]	Shape mask
5	NUM_CLASSES	2(Disease + Background)	Classes number
6	DETECTION_MIN_CONFIDENCE	0.9	Confidence detection
7	POOL_SIZE	7	Pooling size
8	STEPS_PER_EPOCH	500	Number of trains
9	VALIDATION_STEPS	10	Number of validation/ test samples



(a) An original image (b) Annotated image (c) Objects masked by R-CNN

FIGURE 6. The process of the Mask R-CNN



(a) X_i and Y_i of the objects (b) Bounding boxes of the objects

FIGURE 7. Example of cropping position and bounding boxes

Calculate the X and Y positions of circles: The X and Y positions of each circle image consist of four points as shown in Figure 7(a): X_1 and Y_1 are in the upper left corner of the circle, and X_2 and Y_2 are in the bottom right corner of the circle.

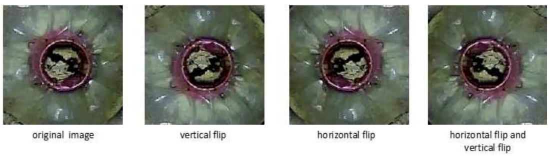
Create bounding boxes from X and Y positions: This step draws a square box from the X and Y positions to illustrate the boundaries of the image as shown in Figure 7(b). The figure shows that the square boxes are drawn precisely in the center of the mushroom cultivation materials.

Calculate the center of the image for cropping: The equation for calculating the center of the image (x_i, y_i) can be calculated from

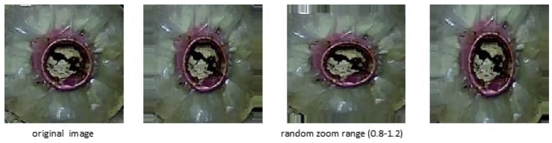
$$x_i = \frac{|\text{distance}(x_1, x_2)|}{2}, \quad y_i = \frac{|\text{distance}(y_1, y_2)|}{2} \tag{1}$$

i is any circle of the image. The cropped image size is 200×200 pixels, calculated from the center of the image as shown in Figure 8. First, all cropped images are divided into two sets by manual classification: a set of diseased images and a set of disease-free images with the same number of images equal to 1,000. Then, these images are processed to find more accurate prognostic models using CNN models.

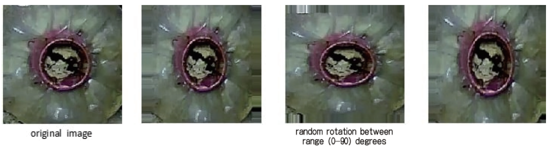
Figure 8. Examples of an image obtained after cropping



(a) Horizontal and vertical flip
It reverses the image vertically and horizontally.



(b) Zoom
Use a value between (0.8-1.2), the normal level is 1.



(c) Rotation
Set to 90 will randomly rotate between (0-90) degrees.

(d) Shear
This image tilt set to 20 will randomly tilt between (0-20) degrees.

(e) Brightness
Adjust the picture brightness value between 0-2
(0 = no brightness and 2 = maximum brightness).

Figure 9. Examples of image augmentation techniques

4) Image pre-processing. This procedure provides image data to evaluate the appropriate CNN model for mushroom prognosis. The experiments divided the image data into two groups: 1) fungal disease and non-fungal disease images applying augmentation techniques [21] and 2) fungal disease and non-fungal images without augmentation techniques. The augmentation techniques increase the variety of images used in the experiment. In addition, they have reduced the overfitting [22] of the data. For this research, we have applied five augmentation methods as illustrated in Figure 9: (a) Horizontal and vertical flip, (b) Zoom, (c) Rotation, (d) Shear, and (e) Brightness.

The experiment's image data is shown in Table 2.

Table 2. The experimental dataset

Data	Training	Validation	Testing	Total
Fungal disease	520 (65%)	120 (15%)	200 (20%)	1,000
Non-fungal disease	520 (65%)	120 (15%)	200 (20%)	1,000

5-6) CNN models. This section presents high-precision CNN techniques to create learning models for mushroom fungal prognosis, including DenseNet201, ResNet50, InceptionV3, and VGG19 [2]. All models are developed in Python 3 by working with Keras v2.6 and Tensorflow v2.6 over Google Colaboratory. These models are assigned the same number of test epochs as 30; the detailed modeling steps are below.

(1) Separate augmentation image dataset and unaugmentation image dataset to compare forecast performance against CNN models.

(2) Define parameters to test the CNN models shown in Table 3.

TABLE 3. Setting parameters running CNN model

No.	Parameters	Value	Remark
1	IMAGE_SIZE	224×224	Images resized to fit the model
2	BATCH_SIZE	32	Size of the dataset to test the models for each epoch
3	MODEL_ EPOCH	300	Number of tests
4	ACTIVATION_FUNCTION	SIGMOID	It is suitable for two classes with data in 0, 1.
5	OPTIMIZERS	SGD	Because we need to update parameters for every training dataset, SGD is a fast algorithm.
6	LOSS FUNCTION	BINARY_CROSSENTROPY	There are only two classes in this research.

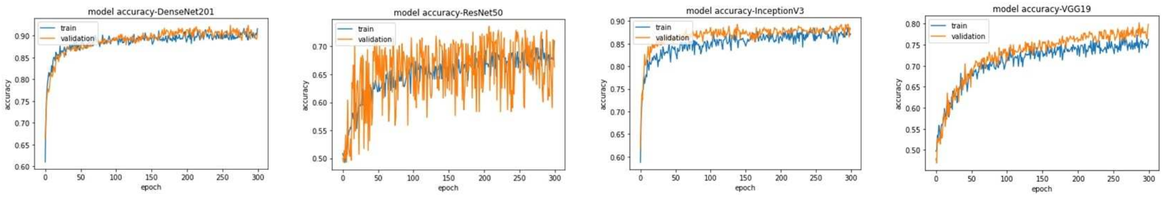
(3) Last, all CNN models are tested against the specified parameters to determine which model has the highest accuracy.

(4) Compare the accuracy of each model and select the most accurate model for further fungal analysis.

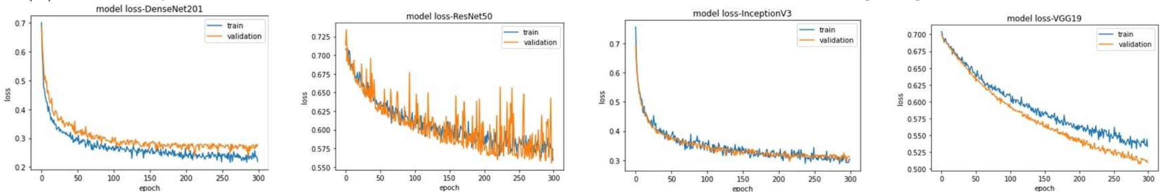
Figure 10 shows the comparison of learning models between DenseNet201, ResNet50, InceptionV3, and VGG19. The results show that DenseNet201 has the highest learning accuracy for both augmentation and non-augmentation, and it also has the lowest overfitting.

From Table 4, after testing all CNN models, the test results show that all models have more than 50% accuracy, especially DenseNet201 has the highest accuracy (augmentation = 89.74% and non-augmentation = 86.50%). Other models have forecast accuracy in the following order: InceptionV3 (aug = 87.25%, non-aug = 77.74%), VGG19 (aug = 83.49%, non-aug = 76.24%), and ResNet50 (aug = 74.25%, non-aug = 70.74%), respectively. Therefore, the DenseNet201 is best suited for the prognosis of fungal disease in our smart mushroom farms.

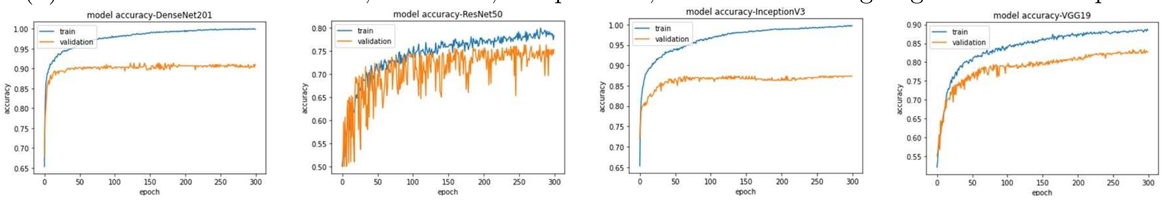
7) Real-time prediction. The DenseNet201 is the most accurate. Therefore, we used this model for real-time (Fungal disease) prediction in the intelligent farm, as shown in Figure 11. The figure consists of three steps: image preparation, Mask R-CNN, and predicting and alerting process. The image preparation process involves getting the path of the image recorded on the computer by the robot and sending one image at a time to the Mask R-CNN process. The images sent from the data preparation process are cropped one at a time with Mask R-CNN. The cropped image has a fixed size of 200×200 pixels transferred to the prognostic process by DenseNet201. Prognostic results are expressed



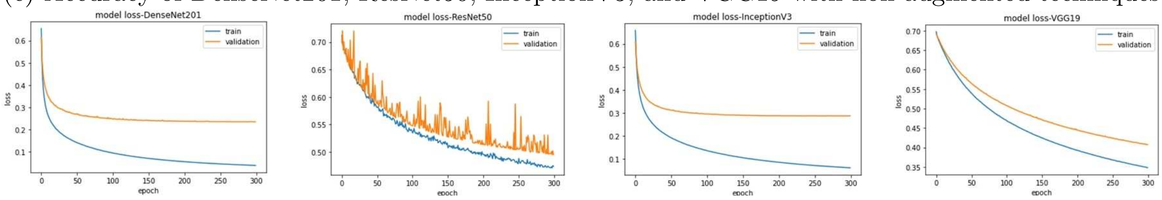
(a) Accuracy of DenseNet201, ResNet50, InceptionV3, and VGG19 using augmented techniques



(b) Loss rate of DenseNet201, ResNet50, InceptionV3, and VGG19 using augmented techniques



(c) Accuracy of DenseNet201, ResNet50, InceptionV3, and VGG19 with non-augmented techniques



(d) Loss rate of DenseNet201, ResNet50, InceptionV3, and VGG19 with non-augmented techniques

FIGURE 10. The comparison of the training model CNNs

TABLE 4. CNNs performance comparison

Testing CNN model	**Class**	**Non-augmentation (%)**				**Augmentation (%)**			
		Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
DenseNet201	Disease	86.50	86.50	88.15	87.45	89.74	91.45	90.25	91.48
	Non-Disease		87.24	85.45	86.50		90.57	92.70	91.74
ResNet50	Disease	70.74	73.72	66.70	69.71	74.25	74.20	75.25	75.23
	Non-Disease		69.01	76.33	72.74		75.24	74.27	74.25
InceptionV3	Disease	77.74	88.74	65.57	74.01	87.25	88.25	85.20	87.26
	Non-Disease		72.71	90.72	80.75		86.23	88.15	87.25
VGG19	Disease	76.24	74.33	81.02	77.67	83.49	81.41	86.49	84.45
	Non-Disease		79.67	72.33	75.24		86.42	80.40	83.49

as a percentage (%) of fungal incidence. If the forecast result exceeds 50% (The default value indicates the fungal disease), the system will immediately alert the farmer. However, this value can be adjusted according to the situation. All three processes are real-time. Therefore, the total time per cycle is 9 minutes, from taking pictures from the robot to alerting via a mobile application (LINE). Each cycle is processed as a square wave with a travel distance of one meter.

4. **Experimental Results and Evaluation.** The evaluation of the effectiveness of this paper is divided into three parts.

1) Mask R-CNN performance evaluation. Intersection over Union or IoU is commonly used for evaluating Mask R-CNN performance. It indicates the accuracy value for measuring the consistency or overlap of an object image. If IoU is greater than or

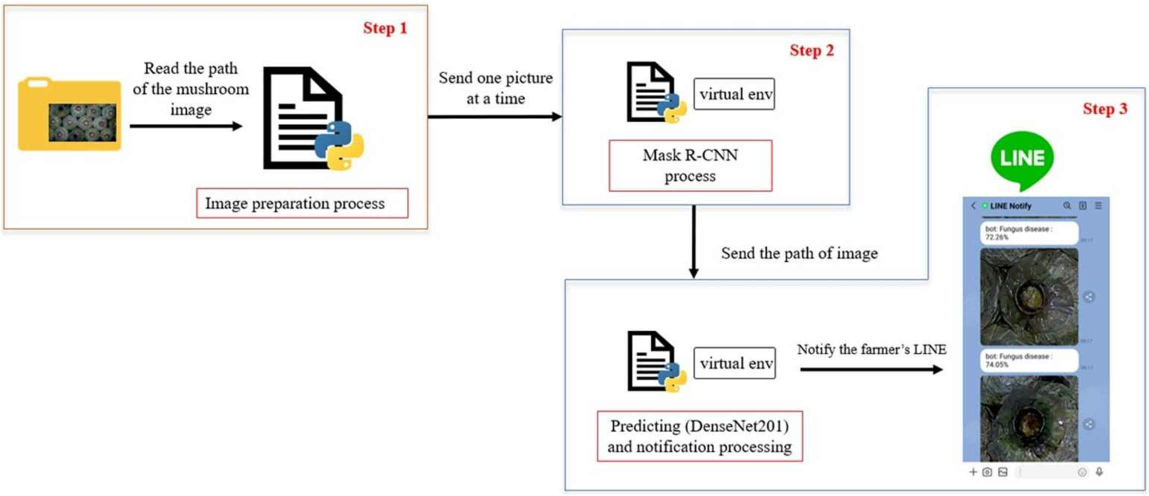
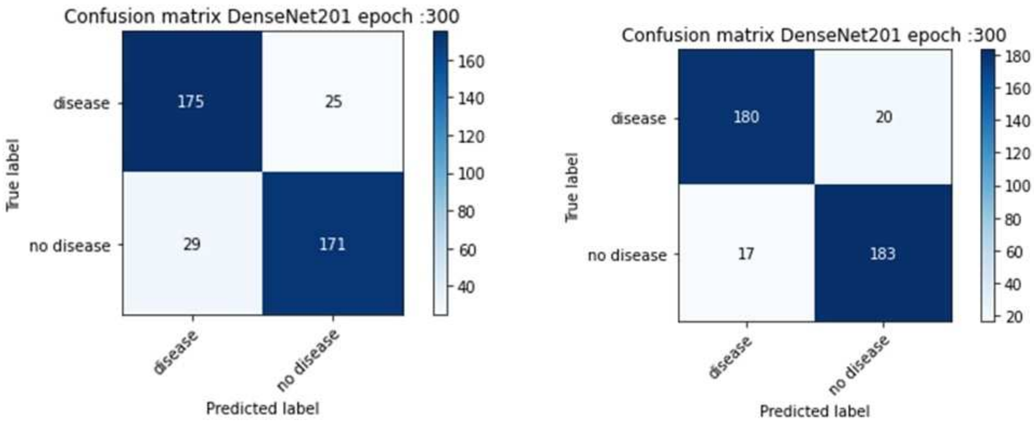


FIGURE 11. Real-time prediction of fungal disease



(a) Augmentation (b) Non-augmentation

FIGURE 12. Performance measurement of CNN model

equal to 0.5 (50%) [20], it is an acceptable overlap. The results showed that the mean consistency of the dataset used in this paper was 0.86 (86%), which is a highly efficient consistency value.

2) CNN model performance evaluation. Of the four CNN benchmarks, we found that the DenseNet201 was the most accurate for the prognosis of fungi disease, as shown in Table 4 in sections 5-6) CNN models. In addition, according to the confusion matrix in Figure 12, the model DenseNet201 using the augmentation technique could analyze fungal disease images more efficiently than the non-augmentation technique.

3) Real-time overall system performance evaluation. According to the real-time forecasting test, each epoch uses several images equal to 100 images; the confusion matrix is shown in Figure 13 and summarized in Table 5. We used the F1 to measure performance because of the efficacy of the predictive model that accurately analyzed the actual disease that was found to be a true disease. At the same time, we also need to detect disease. Therefore, the measurement of the selected model has performance of detecting disease as 73.05%.

From Figure 13, the confusion matrix of real-time diction can summarize diseased fungal images of fairy mushrooms with the following values: accuracy, precision, recall, and F1, as in Table 5.

		Predicted	
		Disease	No disease
Actual	Disease	50	10
	No disease	17	23

FIGURE 13. Confusion matrix of real-time diction

TABLE 5. Confusion matrix of real-time diction

No.	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Disease	73.05	74.62	83.33	78.79
No disease	27.03	69.69	57.50	63.28

5. **Conclusions.** Fungal diseases do not only affect mushrooms' products but also other plants. Moreover, the fungi from the mushroom affect the farmer's health because they will inhale it when they stay in the greenhouse. Thus, the fungal disease analysis of mushrooms (Case study: Fairy mushroom) can solve their problems. This research uses the deep learning technique and image processing technique specified CNN for generating the model. This model uses to learn the characteristics of the fungal diseases of fairy mushrooms. The Mask R-CNN technique is selected to detect the object in an image of the plant position of the mushroom cultivation in a plastic bag and detect the fungal diseases. The results showed that DenseNet201 has the highest accuracy and lowest overfitting. Therefore, it was chosen to predict fungal infections within intelligent farms. Our proposed prototype is predicted in real time. The predictive accuracy during fungal disease is 73.05%, and the system can detect fungal disease in only six hours. As the fungal disease spreads, the system will immediately alert farmers via smartphones via the LINE application. It clearly shows that our proposed system is effective according to our hypothesis. We plan to optimize the forecasting accuracy to be higher than 90% in the future.

REFERENCES

[1] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali and M. Guizani, A survey of machine and deep learning methods for Internet of Things (IoT) security, *IEEE Communications Surveys & Tutorials*, vol.22, no.3, pp.1646-1685, DOI: 10.1109/COMST.2020.2988293, 2020.

[2] J. Chen, J. Chen, D. Zhang, Y. Sun and Y. A. Nanehkaran, Using deep transfer learning for image-based plant disease identification, *Computers and Electronics in Agriculture*, vol.173, DOI: 10.1016/j.compag.2020.105393, 2020.

[3] K. P. Ferentinos, Deep learning models for plant disease detection and diagnosis, *Computers and Electronics in Agriculture*, pp.145311-145318, 2018.

[4] S. P. Mohanty, D. P. Hughes and M. Salathé, Using deep learning for image-based plant disease detection, *Frontiers in Plant Science*, vol.7, DOI: 10.3389/fpls.2016.01419, 2016.

[5] J. Amara, B. Bouaziz and A. Algergawy, A deep learning-based approach for banana leaf diseases classification, *Datenbanksysteme für Business, Technologie und Web (BTW2017)*, vol.266, pp.79-88, 2017.

[6] D. Hughes and M. Salathé, An open access repository of images on plant health to enable the development of mobile disease diagnostics, *arXiv Preprint*, arXiv: 151108060, 2015.

[7] N. Kitpo, Y. Kugai, M. Inoue, T. Yokemura and S. Satomura, Internet of Things for greenhouse monitoring system using deep learning and bot notification services, *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp.1-4, 2019.

[8] S. Khummanee, S. Wiangsamut, P. Sorntepa and C. Jaiboon, Automated smart farming for orchids with the Internet of Things and fuzzy logic, *2018 International Conference on Information Technology (InCIT)*, pp.1-6, 2018.
[9] S. Wiangsamut, P. Chomphuwiset and S. Khummanee, Chatting with plants (orchids) in automated smart farming using IoT, fuzzy logic and chatbot, *Advances in Science, Technology and Engineering Systems Journal*, vol.4, no.5, pp.163-173, 2019.
[10] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, Cambridge, 2016.
[11] G. L. Grinblat, L. C. Uzal, M. G. Larese and P. M. Granitto, Deep learning for plant identification using vein morphological patterns, *Computers and Electronics in Agriculture*, pp.127418-127424, 2016.
[12] X.-B. Jin, X.-H. Yu, X.-Y. Wang, Y.-T. Bai, T.-L. Su and J.-L. Kong, Deep learning predictor for sustainable precision agriculture based on Internet of Things system, *Sustainability*, vol.12, no.4, DOI: 10.3390/su12041433, 2020.
[13] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert and D. Van Valen, Deep learning for cellular image analysis, *Nature Methods*, pp.1-14, 2019.
[14] S. M. Noe, T. T. Zin, P. Tin and I. Kobayashi, Automatic detection and tracking of mounting behavior in cattle using a deep learning-based instance segmentation model, *International Journal of Innovative Computing, Information and Control*, vol.18, no.1, pp.211-220, 2022.
[15] R. Gandhi, *R-CNN, Fast R-CNN, Faster R-CNN, YOLO Object Detection Algorithms*, https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e, 2018.
[16] K. He, G. Gkioxari, P. Dollár and R. Girshick, Mask R-CNN, *2017 IEEE International Conference on Computer Vision (ICCV)*, pp.2980-2988, DOI: 10.1109/ICCV.2017.322, 2017.
[17] Z. Cai and N. Vasconcelos, Cascade R-CNN: Delving into high quality object detection, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.6154-6162, DOI: 10.1109/CVPR.2018.00644, 2018.
[18] J. Yan, H. Wang, M. Yan, W. Diao, X. Sun and H. Li, IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery, *Remote Sensing*, vol.11, no.3, DOI: 10.3390/rs11030286, 2019.
[19] M. Wu, H. Yue, J. Wang, Y. Huang, M. Liu, Y. Jiang et al., Object detection based on RGC mask R-CNN, *IET Image Processing*, vol.14, no.8, pp.1502-1508, 2020.
[20] M. Everingham, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, The pascal Visual Object Classes (VOC) challenge, *International Journal of Computer Vision*, vol.88, no.2, pp.303-338, 2010.
[21] C. Shorten and T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, *Journal of Big Data*, vol.6, no.1, pp.1-48, 2019.
[22] L. Rice, E. Wong and Z. Kolter, Overfitting in adversarially robust deep learning, *International Conference on Machine Learning*, vol.119, pp.8093-8104, 2020.