International Journal of Advanced Smart Convergence Vol.13 No.1 212-220 (2024)
http://dx.doi.org/10.7236/IJASC.2024.13.1.212

IJASC 24-1-24

Accuracy Measurement of Image Processing-Based Artificial Intelligence Models

Jong-Hyun Lee[1], Sang-Hyun Lee[2]

[1]*ESHEL Tree Co., Gwangju, Korea*
[2]*Associate Professor, Department of Computer Engineering, Honam University, Korea*
saintvk@naver.com [1], leesang64@honam.ac.kr[2]

Abstract

When a typhoon or natural disaster occurs, a significant number of orchard fruits fall. This has a great impact on the income of farmers. In this paper, we introduce an AI-based method to enhance low-quality raw images. Specifically, we focus on apple images, which are being used as AI training data. In this paper, we utilize both a basic program and an artificial intelligence model to conduct a general image process that determines the number of apples in an apple tree image. Our objective is to evaluate high and low performance based on the close proximity of the result to the actual number. The artificial intelligence models utilized in this study include the Convolutional Neural Network (CNN), VGG16, and RandomForest models, as well as a model utilizing traditional image processing techniques. The study found that 49 red apple fruits out of a total of 87 were identified in the apple tree image, resulting in a 62% hit rate after the general image process. The VGG16 model identified 61, corresponding to 88%, while the RandomForest model identified 32, corresponding to 83%. The CNN model identified 54, resulting in a 95% confirmation rate. Therefore, we aim to select an artificial intelligence model with outstanding performance and use a real-time object separation method employing artificial function and image processing techniques to identify orchard fruits. This application can notably enhance the income and convenience of orchard farmers.

Keywords: *Machine learning, Deep learning, CNN models, Random forest models, VGG16 models, Image processing*

1. Introduction

Recently, research has been actively conducted in the field of image processing to enable efficient management by combining object detection and tracking technology of intelligent surveillance systems with various services, products, and industries. In many fields of industry, systems that can detect objects in real time using image processing technology are used [1], but as the number of objects to be managed increases, it is difficult and expensive to detect them, so there are practical difficulties [2].

Therefore, it is necessary to apply an artificial intelligence model that can efficiently detect even if the

Manuscript Received: February. 6, 2024 / Revised: February. 15, 2024 / Accepted: February. 26, 2024
Corresponding Author: leesang64@honam.ac.kr
Tel: +82-62-940-5285, Fax: +82-62-940-5285
Associate Professor, Department of Computer Engineering, Honam University, Korea

number of objects is large at a low cost, and it is essential to conduct research to improve the performance of the existing system so that it can detect objects individually. A system that can detect and track objects using artificial intelligence is required to be used in various environments as it is used in various fields, while the more numerous and crowded the environment, the more it fails to detect individual objects in the image, so it can no longer detect and track objects, and the performance is low.

In particular, a typical object tracking algorithm can detect and track individual objects when they are separated, but in some cases, the target object is lost or mistaken for another object during the tracking [3]. An important cause of this object tracking failure is the object detection error, in which two or more different objects are detected as one object without the boundary distinction between the objects when they are in close proximity. In order to improve object detection errors and improve the performance of AI detection systems, it is necessary to have a way to accurately separate proximity objects individually in real time. Therefore, in the field of image processing, research for object separation is continuously attempted. By using this image processing system, it is increasing the convenience of a wide range of industries, and it is helping a lot in the current situation where there is a shortage of labor in rural areas.

A representative research related to this is a study on a system that uses unmanned robots to detect and harvest fruits, and it suggests how to realize the automation of agricultural work by combining computer vision and robotic technology [4]. The study aims to develop a system for accurately detecting fruit and calculating yields in an orchard environment as a technology for real-time fruit detection and counting, providing a comprehensive overview of automated fruit detection technology in orchards and its potential applications in agriculture using various sensors and image processing technologies [5]. As such, it deals with research on fruit detection technology, and studies how to detect and manage fruit using various image processing and deep learning technologies.

Currently, natural disasters are severely impacting agricultural production and farmers' incomes. In particular, natural disasters such as typhoons cause fruit to fall in many orchards, resulting in a decrease in yields and income for farmers.

To address these issues, fruit crop accident insurance is a type of insurance policy that protects farmers from crop loss, providing special protection for fruit crops. Premiums and compensation can vary depending on a variety of factors, mainly considering the type of crop, the area where it is grown, and the amount insured. In the case of a natural disaster, the damage assessor visits the fruit farmer to check the details of the fall of the fruit and then compensates. These methods are carried out very slowly, and the method of identifying fallen fruits is also carried out in an unscientific way, which makes it difficult to manage. In order to solve these problems, various artificial intelligence models have been applied to make important advances in the field of object detection and tracking, and they are used in various applications.

The method proposed in this paper proposes a method to accurately identify fallen fruits in orchards using various artificial intelligence models and traditional image processing methods (TIPM). The AI models used are VGG16 models, RandomForest models, and CNN models to estimate the exact number of red apple fruits open in an image of an apple tree, and to evaluate the performance of each model.

2. Related Research

Image processing is an important area for all fields related to image generation, interpretation, and recognition using computers, and was initially mainly used for diagram recognition in image formats, and there are methods such as edge detection [6] and Huff transformation [7] to recognize simple straight objects in diagrams or to extract regions of interest or denoise using morphology operations [8]. Figure 1 shows a typical image processing process.

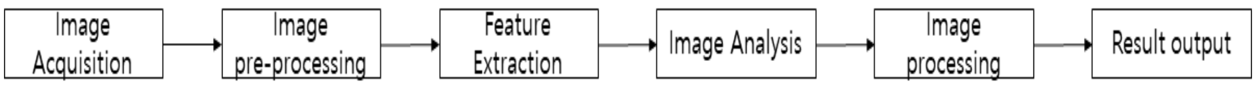

Figure 1. General sequence of image processing

Recently, with the advent of deep learning development tools such as Tensorflow, Keras, and PyTorch, the adoption of GPUs and the improvement of parallel computation capabilities have greatly reduced the training time of deep learning models(DLM). Due to these changes in the environment, deep learning-based research is actively being conducted in various fields, and in particular, deep learning technology based on CNN is attracting great attention in the field of image recognition.

Various artificial intelligence models such as the VGG16 model, the RandomForest model, and the CNN model used in this paper are used to perform image processing to detect objects in the image. These models can recognize different classes and modified objects within an image if enough training data is provided. Here, the VGG16 model is widely used for image classification and feature extraction, and it learns various features in an image, classifies an image based on it, detects a specific object, takes the image as input, and extracts the characteristics of the image by passing through various layers, which is useful in image processing tasks. The RandomForest model is an ensemble of machine learning techniques used for image classification and object detection. It combines multiple decision trees to learn various patterns and features within an image, classify the input image, or detect objects. These RandomForest models can be used for a variety of image processing tasks, including image classification, object detection, and denoising to provide high accuracy and reliability. Therefore, the VGG16 model and the RandomForest model used in this paper can be used as an important tool in image processing and can contribute to the extraction and processing of various information in the image.

3. Proposed AI Model Design

In this study, we propose a method to accurately identify fallen fruits in orchards using artificial intelligence models and traditional image processing methods. The AI models used are VGG16 models, RandomForest models, and CNN models to estimate the exact number of red apple fruits open in an image of an apple tree, and to evaluate the performance of each model. As a performance measurement method for each model, the exact number of red apple fruits open on the image of an apple tree was estimated. The exact number of red apples was followed by a process shown in Figure 2 as a method of estimating the number of fruits.

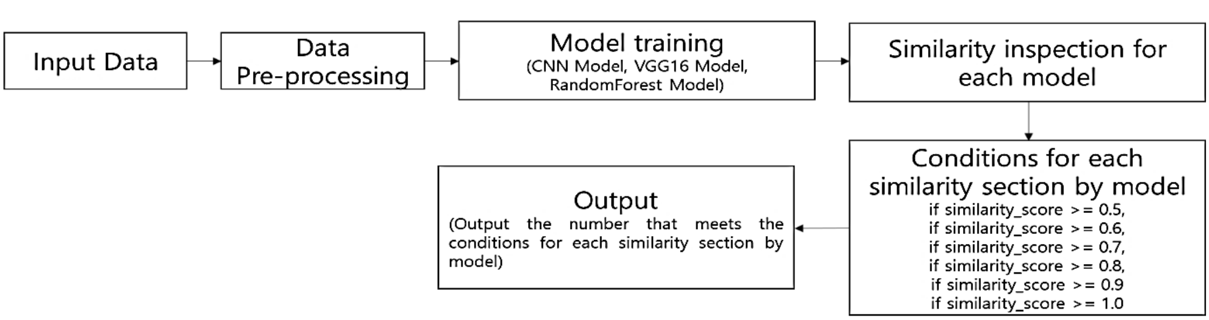

Figure 2. Process of model training and performance evaluation

As shown in Figure 2, the exact number of red apple fruits was estimated in 6 intervals with the condition value, and the similarity of the shape of the apple in the same shape as the learned red apple model was measured at 50%, 60%, 70%, 80%, 90%, and 100%, respectively.

The CNN model is designed as shown in Figure 3. First, the input image starts with a 224x224 pixel size and has three channels (RGB).

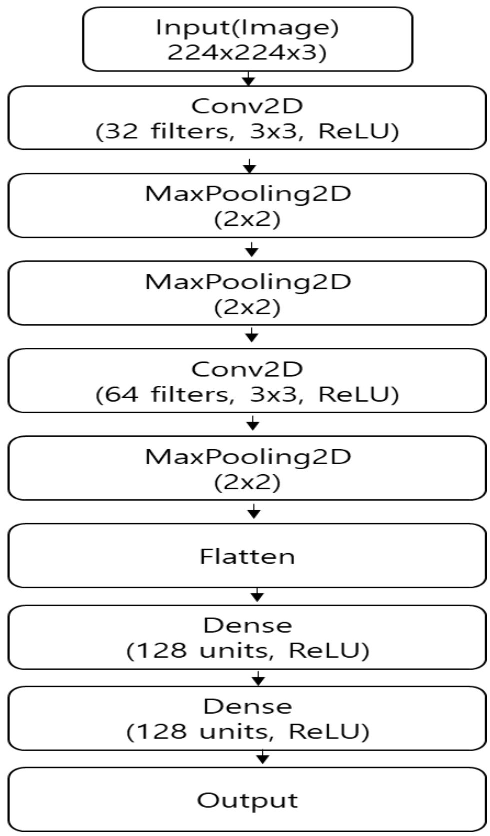

Figure 3. Structural Design of CNN Models

The first convolutional layer uses 32 3x3 filters and ReLU as the activation function. The first MaxPooling layer downsamples the image to a 2x2 size. The second convolutional layer uses 64 3x3 filters and Recitified Linear Unit (ReLU) as the activation function. The second MaxPooling layer downsamples the image back to a 2x2 size. The Flatten layer flattens the output from the previous layer and converts it into a one-dimensional vector. Fully Connected Layer 1 has 128 neurons and uses ReLU as an activation function. The output layer has 1 neuron for binary classification and uses a Sigmoid as an activation function. CNN models consisting of these structures are used to extract features from images and perform classification tasks. The Convolutional layer detects the spatial features of the image, while the MaxPooling layer downsamples the image to increase computational efficiency. The Fully Connected layer performs the final classification based on the extracted features.

The Random Forest model is an ensemble-based machine learning model that uses multiple decision trees to make predictions. In the given code, we're using the RandomForestClassifier class to create and set up a

Random Forest model.

```
# RandomForest model creation
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

RandomForestClassifier class: This class is for generating a Random Forest model provided by the Scikit-Learn library. Random Forest can be used for both Classification and Regression problems, and we use this class to address classification problems. n_estimators=100: This parameter specifies the number of Decision Trees that make up the Random Forest model. In the given code, we are set to use 100 decision trees. More decision trees can improve the stability and performance of the model, but it also increases computational costs. random_state=42: This parameter sets the random number occurrence seed. The Random Forest model randomizes the decision tree, so each run generates a different model. Setting up the random_state ensures that the same results can be reproduced. This means that you can get the same results every time you train a model with the same data and settings.

The structure of traditional image processing techniques is designed as shown in Figure 4. Based on the code given in Figure 4, the image processing steps proceed as follows. After loading the original image from the given image file path, apply image filtering Gaussian blur to blur the image. blurred_image = cv2. GaussianBlur(image, (5, 5), 0) is the application code, where cv2. You can use the GaussianBlur() function to blur the image by applying Gaussianblur to remove noise or soften the borders. BGR to HSV conversion method hsv_image = cv2.cvtColor(blurred_image, cv2. COLOR_BGR2HSV) code, where the cv2.cvtColor() function is used to convert the image from the Blue-Green-Red (BGR) color space to the Hue-Saturation-Value (HSV) color space. The range definition method in the red area is divided by lower_red = np.array([0, 100, 100]) # the lower bound of the red range and upper_red = np.array([10, 255, 255]) # the upper bound of the red range. Here, we define a range of red in the HSV image to find the red area, which means specifying the Hue value of the red pixel as a range. The red mask is generated using the code red_mask = cv2.inRange(hsv_image, lower_red, upper_red), where the cv2.inRange() function is used to mask the pixels within the red range. As a morphology transformation (denoising) method, kernel = np.ones((5, 5), np.uint8), morphology_image = cv2.morphologyEx(red_mask, cv2. MORPH_CLOSE, kernel) code, where the cv2.morphologyEx() function is used to apply morphological transformations to the image to remove noise, mainly cv2. Use MORPH_CLOSE to perform a close operation. This process results in image preprocessing, which involves filtering the image, identifying red areas, and applying morphological transformations to the red mask to remove noise.

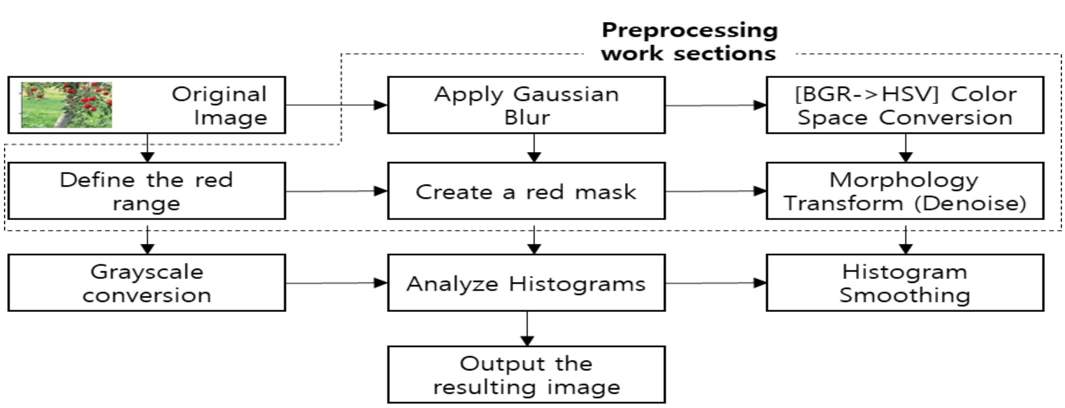

Figure 4. Structural Design of General Image Processing

As shown in Figure 5, the resulting image of each step is displayed as a single figure and output.

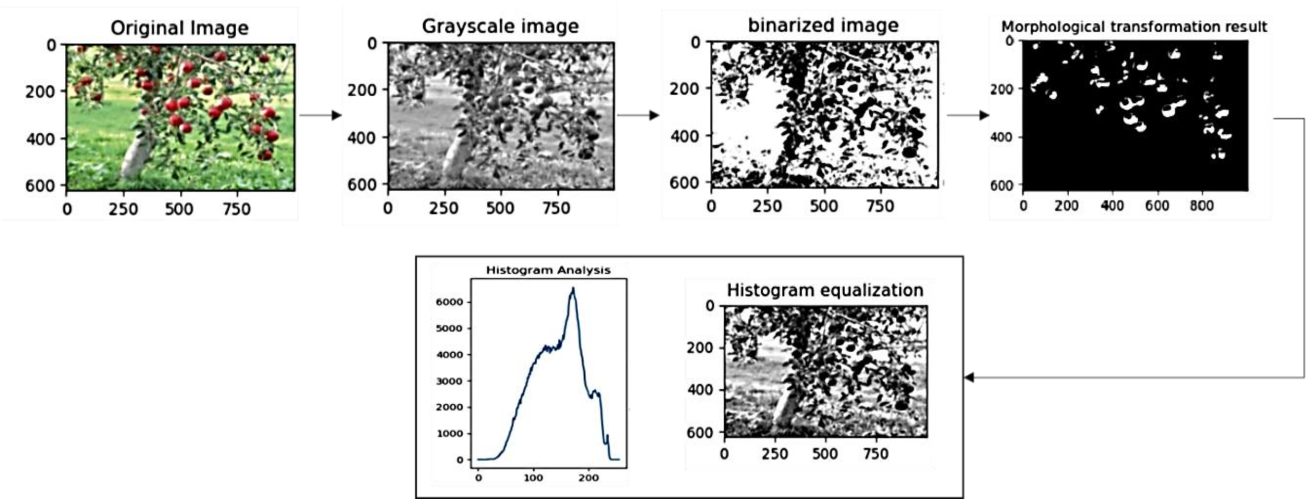

Figure 5. The shape of the result of the image process.

The structure of the VGG16 model shows each layer and its connection structure as shown in Figure 6. To briefly explain the structure of the VGG16 model, it first takes an input layer of 224x224 size color images, and there are 5 Convolutional Layers (Conv), each of which uses a kernel of different sizes to extract the features of the image. It uses ReLU as an activation function, and after the Conv layer, there is a Max Pooling layer, which performs the function of down sampling the image to reduce its size and increase computational efficiency. Fully Connected Layers (FCL) generate high-dimensional feature vectors based on the features of the Conv layer. This vector will be used to perform image classification. Finally, the Output Layer is the output layer for the classification task, which uses the softmax activation function to output the probability for each class.

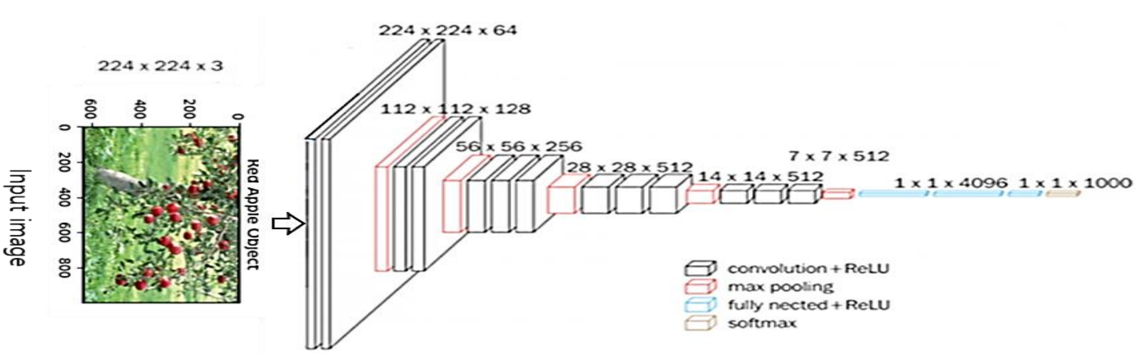

Figure 6. Structural Design of General Image Processing

The VGG16 model consists of a total of 16 layers, which are used to classify images and extract features. The model uses pre-trained weights from large image datasets to provide effective feature extraction.

4. Implementation and Results

The proposed approach employs a range of artificial intelligence models and conventional image processing techniques to precisely count the volume of fallen fruits in an orchard. The utilized artificial intelligence models include the VGG16, RandomForest, and CNN models to estimate the precise quantity of open red apples within an image of an apple tree and assess each model's overall performance.

The objective of this study is to assess the efficiency of the proposed models in determining the precise count of red apple fruits in an image of an apple tree located in an apple orchard. In this context, the use of technical term abbreviations will be defined upon their first usage. The text adheres to a conventional structure, with clear, concise sentences and paragraphs that establish causal connections between statements for a coherent and logical flow of information. To this end, we employed artificial intelligence models, namely VGG16, RandomForest, and CNN models, along with traditional image processing methods, and compared and analyzed their performance. The paper's organization adheres to common academic sections and maintains consistent author and institution formatting, while titles are accurate and comprehensible. The language is objective, formal, and value-neutral, with passive tone and impersonal constructions. The writing uses high-level, standard language and avoids biased, figurative, emotional, or ornamental expressions. Additionally, we followed a consistent citation style and marked quotes to ensure proper formatting features. Finally, the writing avoids filler words, slang, and contractions, and is free from grammatical errors, spelling mistakes, and punctuation errors.

4.1 DataSet

To evaluate the AI model, we organized the dataset into apple images, apple tree leaves, and apple tree ste ms as shown in Table 1. Of the 352 total image data used in this paper, 292 apple images, 30 apple tree leave s, and 30 apple tree stems were used.

Table 1. Main parameters

Division	Number of apples	Number of tree leaves	Number of tree trunks	Total
Training data	200	10	10	**220**
Test data	42	10	10	**62**
Validation data	50	10	10	**70**
Total	**292**	**30**	**30**	**352**

In this study, the software environment for the experiment was developed using Python 3.10 version, the artificial intelligence library used the PyTorch-based MMDetection API, and the hardware environment was Windows 11 for OS, i9-9900k for CPU, 128GB for RAM, and 128GB for GPU. NVIDIA RTX 6000 was used, and a detailed environment is indicated in Table 2.

Table 2. Configuration of Development Environment.

Division	Specification
operating system (OS)	Window 11
central processing unit (CPU)	Intel i9 9900K
GPU	NVIDIA QUADRO RTX6000
Memory	128GB
Storage	Samsung M.2 1TB

4.2 Result

The method used in this paper is an artificial intelligence model, using the VGG16 model, the RandomForest model, and the CNN model, and traditional image processing methods to estimate the exact number of open red apple fruits in the image of an apple tree, and to compare and evaluate the performance of each model. Table 4 shows the apple object detection ratio by model.

Table 3. Apple detection results by model

Division	Model	Number of original image objects	Correct answer (49 pieces)	Accuracy by similarity score
1	CNN	49	47	98%
2	VGG16	49	61	88%
3	RandomForest,	49	32	83%
4	Classical image processing	49	52	97%

In terms of the accuracy of each model, the CNN model first correctly detected 47 out of 49 objects, regardless of the number of original image objects and similarity scores, resulting in a 98% accuracy rate. The VGG16 model correctly detected 61 out of 49 objects, regardless of the number of original image objects and similarity scores, with an accuracy rate of 88%. The RandomForest model correctly detected 32 out of 49 objects, regardless of the number of original image objects and similarity scores, with an accuracy rate of 83%. Classical Image Processing correctly detected 52 out of 49 objects, regardless of the number of original image objects and similarity scores, with a 97% accuracy rate.

These results allow you to evaluate the performance of each model. The CNN model has excellent results in terms of accuracy, and the VGG16 model also has a high degree of accuracy. The RandomForest model has been shown to be less accurate, while the Classical Image Processing method has a very high accuracy. The choice of model may vary depending on your goals, and you will need to consider things like accuracy and calculation speed.

5. Conclusion

In this study, various AI models and traditional image processing techniques were used to improve the detection of red apple objects in orchards. This allowed them to estimate the exact number of red apples and compare their performance. The results of the study suggest the following important conclusions: The CNN model showed an excellent ability to accurately detect objects even with low similarity scores, detecting red apple objects with 98% accuracy. The model provides high reliability in object detection and analyzes the individual characteristics of objects well.

The VGG16 model is capable of detecting a wide variety of objects, and it detected the red apple object with 88% accuracy. The model is suitable for dealing with a variety of object classes and extracts a variety of features within an image. The RandomForest model had a rather low ability to detect objects, detecting red apple objects with 83% accuracy. The model is based on machine learning algorithms and can be used for common object classification tasks. Classical Image Processing has a high degree of accuracy, detecting the red apple object with 97% accuracy. The model is based on traditional image processing methods and performs well at accurately identifying objects within images. Therefore, this study is expected to examine and compare different models and techniques for the detection of red apple objects in orchards to help select the most appropriate model for a particular situation. Future research will apply these models to real-world orchard environments and explore further advances in building real-time object detection systems. It is expected that this will make an innovative contribution to fruit farmers and crop management.

References

[1] X. Zhu, S. Mukhopadhyay, and H. Kurata, “A Review of RFID Technology and its Managerial Applications in Dif ferent Industries,” Journal of Engineering and Technology Management, Vol. 29, No. 1, pp. 152-167, 2012.

[2] L. Ruiz Garcia, and L. Lunadei, “The Role of RFID in Agriculture: Applications, Limitations and Challenges,” Computers and Electronics in Agriculture, Vol. 79, No. 1, pp. 4250, 2011.

[3] S. Zuo, L. Jin, Y. Chung, and D. Park, “An Index Algorithm for Tracking Pigs in Pigsty,”In P roc. of ICITMS, 2014.

[4] Y. Onishi, Yoshida, T., Kurita, H. et al. “An automated fruit harvesting robot by using deep learning,” Robomech J 6, 13, 2019.
DOI: https://doi.org/10.1186/s40648-019-0141-2

[5] F. Xiao, H. Wang, Y. Xu, R. Zhang, “Fruit Detection and Recognition Based on Deep Learning for Automatic Harvesting: An Overview and Review,” Agronomy, Vol. 13, No. 6, pp. 1625, 2023.
DOI: https://doi.org/10.3390/agronomy13061625

[6] S-. Y. Lee, T.-C. Huynh, J.-H. Park, J.-T. Kim, “Bolt-Loosening Detection using Vision-Based Deep Learning Algorithm and Image Processing Method,” Journal of the Computational Structural Engineering Institute of Korea. pp. 265-272, 2019.
DOI: https://doi.org/10.7734/COSEIK.2019.32.4.265

[7] J.Oh, “Straight Line Detection Using PCA and Hough Transform,” Journal of the Korea Institute of Information a nd Communication Engineering. Vol. 22, No. 2, pp. 227~232, Feb. 2018.

[8] Y-. K. S.-H. Kim, “On the TFT LCD Cell Defect Inspection Algorithm using Morphology,” Journal of the Korean Institute of Illuminating and Electrical Installation Engineers, Vol. 21, No. 1, January 31, 2007.
DOI: https://doi.org/10.5207/jieie.2007.21.1.019

[9] G-. S. Lee, S-. H. Lee, “Prediction of changes in fine dust concentration using LSTM model,” International Journal of Advanced Smart Convergence (IJASC), vol. 11, no.2, 2022.
DOI: https://doi.org/ IJASC.2022.11.2.05

[10] B-. S. Jang, S-. H, Lee, “CNN based Sound Event Detection Method using NMF Preprocessing in Background Noise Environment,” International Journal of Advanced Smart Convergence (IJASC), vol. 9, no.2, 2020.
DOI: https://doi.org/ IJASC.2020.9.2.3