agronomy

MDPI

Article

Feature Extraction for Cocoa Bean Digital Image Classification Prediction for Smart Farming Application

Yudhi Adhitya [1,*], Setya Widyawan Prakosa [2], Mario Köppen [1] and Jenq-Shiou Leu [2]

[1] Department of Computer Science and Systems Engineering (CSSE), Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka-shi, Fukuoka 820-8502, Japan; mkoeppen@ieee.org

[2] Department of Electronic and Computer Engineering (ECE), National Taiwan University of Science and Technology, Taipei City 106, Taiwan; d10702804@mail.ntust.edu.tw (S.W.P.); jsleu@mail.ntust.edu.tw (J.-S.L.)

* Correspondence: adhitya.yudhi898@mail.kyutech.jp

Received: 4 September 2020; Accepted: 5 October 2020; Published: 25 October 2020

Abstract: The implementation of Industry 4.0 emphasizes the capability and competitiveness in agriculture application, which is the essential framework of a country's economy that procures raw materials and resources. Human workers currently employ the traditional assessment method and classification of cocoa beans, which requires a significant amount of time. Advanced agricultural development and procedural operations differ significantly from those of several decades earlier, principally because of technological developments, including sensors, devices, appliances, and information technology. Artificial intelligence, as one of the foremost techniques that revitalized the implementation of Industry 4.0, has extraordinary potential and prospective applications. This study demonstrated a methodology for textural feature analysis on digital images of cocoa beans. The co-occurrence matrix features of the gray level co-occurrence matrix (GLCM) were compared with the convolutional neural network (CNN) method for the feature extraction method. In addition, we applied several classifiers for conclusive assessment and classification to obtain an accuracy performance analysis. Our results showed that using the GLCM texture feature extraction can contribute more reliable results than using CNN feature extraction from the final classification. Our method was implemented through on-site preprocessing within a low-performance computational device. It also helped to foster the use of modern Internet of Things (IoT) technologies among farmers and to increase the security of the food supply chain as a whole.

Keywords: textural feature; co-occurrence matrix; convolutional neural network; image processing; smart farming; artificial intelligence; machine learning

1. Introduction

Information technology has indeed shifted very significantly in human life. It is undeniable that technology currently represents an essential role in the development process from time to time. We are entering the Industrial Revolution Era 4.0, where Internet of Things (IoT) technologies are very influential in everyday life. Even in the area of agriculture, such technologies [1,2] have many important roles. Feature extraction is an artificial intelligence (AI) method that selects or consolidates numerous variables as a feature, which can effectively decrease the substance of data processed while still representing the fundamental dataset. The primary feature extraction for texture analysis, developed around the 1970s, employs co-occurrence matrix features introduced by [3].

Utilization of the feature extraction [4] method has been a common practice for image classification since the development of machine learning schemes. Combining several feature extraction procedures might help to improve the classification accuracy, but it can also overload the hardware, which impacts the duration of computing time. The use and utilization of deep learning in the application of AI schemes were studied by [5]. With the expansion, one of the architectures discussed was convolutional neural network (CNN)-based feature extraction. Utilizing the feature extraction method as part of the classification process was carried out by [6–10].

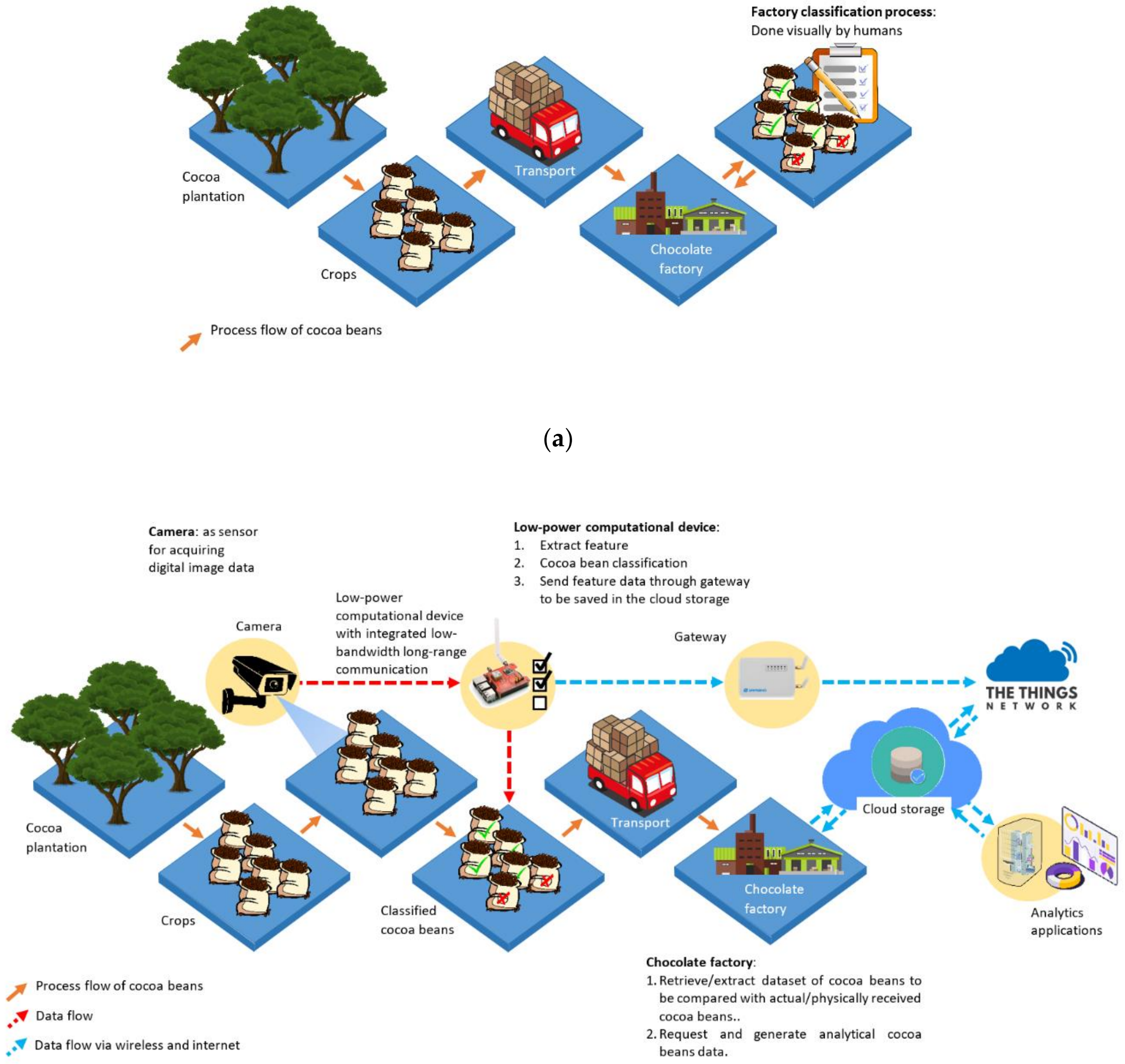
Texture is an essential feature in image processing. One of the characteristics that determines a texture is the distribution of the grayscale's spatial value; therefore, applying statistical features is one of the methods suggested in the various studies on machine vision. The most abundant texture analysis method used is based on first-order and second-order statistical functions of the gray level co-occurrence matrix. Initially, it was introduced as a co-occurrence matrix to extract features for image analysis of satellite imagery [3]. Textural feature analysis using the gray level co-occurrence matrix (GLCM), local binary pattern (LBP), local binary pattern gray level co-occurrence matrix (LBP-GLCM), gray level run length matrix (GLRLM), and segmentation-based fractal texture analysis (SFTA) methods for extracting features from and classification of histopathological images was applied by [11,12]. Application of textural features as a preprocessing parameter was carried out by [13] for magnetic resonance imaging (MRI) images. Research on textural analysis was also conducted by [14], utilizing the GLCM and the function of quantization level [15].

The utilization of technology is necessary to increase agricultural production [16], especially in terms of quality and competitiveness. The availability of technological innovations such as machine learning and deep learning [17–19] is also one of the keys to improving farmer welfare and attracting the younger generation's interest in creating various derivative business opportunities.

Several researchers [20,21] have conducted the identification and grading systems, including the classification of crops [22]. They utilized image processing technology and a support vector machine (SVM) classifier to determine the degree of fermentation to meet quality control requirements [23], and implementation of an artificial neural network (ANN) structure with 35 input nodes for classifying four cocoa beans classes, namely, *whole beans*, *broken beans*, *bean fractions*, and *skin-damaged beans*. The manipulation of histograms for texture classification was carried out by [24] using the sum and difference histogram technique to apply the co-occurrence matrix in texture analysis. The application of deep learning techniques was carried out by [25] to classify fruit based on species class and price in supermarkets. Moreover, machine learning techniques have also been applied in automation in agriculture [26]. In [27], a study was conducted on the fermentation rate of cocoa beans to make it easier for farmers to measure the fermentation rate and to ensure the quality of cocoa beans.

Sorting activity is a designated stage to determining the quality and classifying the grade of cocoa beans based on their physical size (quality level) and, at the same time, to separating the miscellaneous dirt contained in it. The manual sorting and classification process has recently been used to classify beans based on their physical appearance and size. During the classification process, the processing factory can separate the good beans from the regular ones to obtain a more suitable payment for the farmer on a higher-quality grade [28]. This conceptual framework can be seen in (Figure 1a). However, the final quality check has to be conducted at the factory site when the beans arrive.

Smart farming signifies an example of information technology advancement [29], empowering farmers to apply more dependable control. Reducing time-consuming data accumulation can promptly be accomplished through real-time, on-site processing, and can later be transmitted through the cloud for further analysis [1]. The adoption of sensor technology [2,30] to consolidate hybrid data with the implementation of network technology that can reach remote areas and a wide coverage area [31,32], and which can be sustained by cloud data storage [33,34], is advantageous to farmers for enhancing and maximizing their productivity and to comprehend a smart farming system. Data stored within the cloud can also be used by processing plants to resolve operational management problems.


Factory classification process:
Done visually by humans
Cocoa plantation
Crops
Transport
Chocolate factory
Process flow of cocoa beans
Camera: as sensor for acquiring digital image data
Low-power computational device:
1. Extract feature
2. Cocoa bean classification
3. Send feature data through gateway to be saved in the cloud storage
Low-power computational device with integrated low-bandwidth long-range communication
Camera
Gateway
THE THINGS NETWORK
Cloud storage
Classified cocoa beans
Analytics applications
Chocolate factory:
1. Retrieve/extract dataset of cocoa beans to be compared with actual/physically received cocoa beans..
2. Request and generate analytical cocoa beans data.
Data flow
Data flow via wireless and internet

(**a**)

(**b**)

Figure 1. Process flow of cocoa beans from the cocoa plantation site to the processing factory. (**a**) Traditional concept for the cocoa bean classification framework. (**b**) Smart farming framework concept of cocoa bean classification.

This study investigated a systematic approach to detect classification types of cocoa beans based on industry standards for processing cocoa beans. This approach utilizes digital images of cocoa beans to expedite their assessment and classification, which currently requires manual assessment and classification by humans (Figure 1a).

We designed a smart farming framework concept scheme (Figure 1b) to explain our requirements and the use of our research methodology to address numerous references. Located in remote area, this framework utilizes the camera as a sensor node to capture cocoa bean images to be analyzed. Next, for local processing, employing a low-power computational device, integrated and connected to the low-bandwidth long-range communication network, the classification process can be accomplished. The data are later transmitted to cloud storage to be further analyzed in real-time by the processing factory.

The implementation expectations of this proposed approach a decrease in the processing time for assessment and classification analysis, which results in a reduction in time for the further processing of cocoa beans. Our proposal consists of utilizing the co-occurrence matrix features of the GLCM and CNN for feature extraction.

2. Development of a New Methodology for the Textural Feature Analysis of Digital Images of Cocoa Beans

Image feature extraction is one of the steps of extracting data from objects in an image to distinguish them from other objects. The extracted features are then used as parameter values to distinguish between objects as a part of the classification stages.

The GLCM is an example of the basic and most traditional method of feature extraction and employs first- or second-order statistical characteristics. Toward the evolution of machine learning, CNN also utilizes the feature extraction process.

There are several steps for classifying cocoa beans: the first is image capturing, that is, obtaining images of cocoa beans and storing them in electronic file form. The second is preprocessing, which involves eliminating unnecessary parts of an image. Next is the feature extraction; within this study, we adopted the GLCM and CNN methods. From numerous steps of the classification process, the final result is the classification of the cocoa beans.

The classification of cocoa beans is used as the basis for processing cocoa beans to achieve the best results in the finished products. The final result of the classification process of cocoa beans can later be stored in the cloud, as cloud storage can store or archive data easily, so as to prevent data from being damaged or lost when stored on local storage media, such as hard disks or flash drives.

3. Case Study and Results

3.1. Materials and Methods

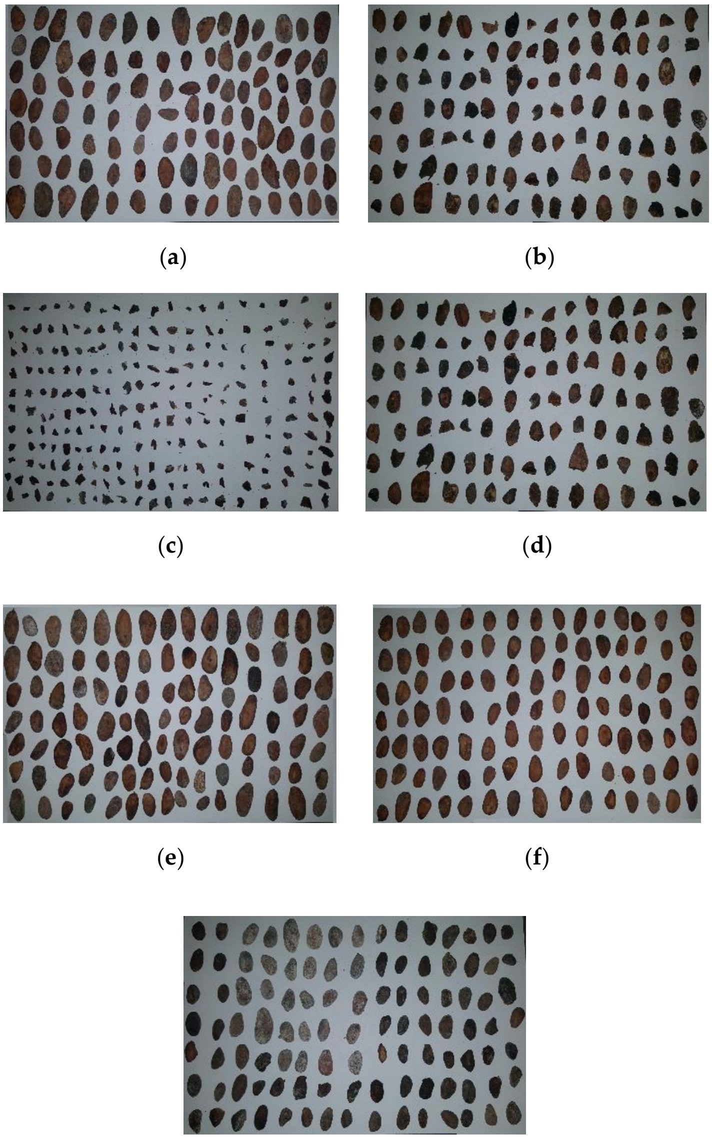
3.1.1. Materials

The digital images of the cocoa beans for the study were procured from South Sulawesi, Indonesia. The sampling method was based on [35,36]. These cocoa bean samples were classified as the following: (1) *whole beans*, cocoa beans with a whole seed coat covering all of the seed parts and not showing any fracture (Figure 2a); (2) *broken beans*, a cocoa bean with a missing portion that is half (1/2) or less than the whole cocoa bean (Figure 2b); (3) *beans fractions*, a cocoa bean fraction that is less than half (1/2) of the whole cocoa bean (Figure 2c); (4) *skin-damaged beans*, a cocoa bean with a missing bean shell that is half or less in size than the whole cocoa bean (Figure 2d); (5) *fermented beans*, a cocoa bean, which is the final product of the curing process and that is washed or unwashed and dried (Figure 2e); (6) *unfermented beans*, half or more of the sliced grayish chips' surface is visible in cocoa beans (e.g., slate or solid grayish blue in color and texture), while the surface is dirty white (Figure 2f); (7) *moldy beans*, a cocoa bean with mold on the inside, and when the cocoa beans are split open, the fungus can be seen with the eye (Figure 2g). The cocoa bean sample classification utilized the Indonesian Standardization Institution's base classification as a regulation reference to export Indonesian cocoa beans [37].

These cocoa bean digital images were collected at the factory, and the final goal was to help reduce the classification process at the factory site. The cocoa bean image acquisition was achieved using a compact digital camera, as depicted in Figure 3.

The digital image dataset of the cocoa beans contained seven classes of cocoa bean classifications. Among all of the 7428 images, 1187 were listed as *whole beans*, 1046 as *broken beans*, 426 as *bean fractions*, 822 as *skin-damaged beans*, 916 as *fermented beans*, 1776 as *unfermented beans*, and 1255 as *moldy beans* (see Figure 4).

We split the dataset into 75% for training, 10% for testing, and 15% for validation to conduct the research and analysis. The small visual appearance of the cocoa beans and the rather large number of relevant classes made it necessary to use rather large shares of the material for training.



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

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

(**e**) (**f**)

(**g**)

Figure 2. Cocoa bean sample images from the seven classes: (**a**) *whole beans*; (**b**) *broken beans*; (**c**) *bean fractions*; (**d**) *skin-damaged beans*; (**e**) *fermented beans*; (**f**) *unfermented beans*; (**g**) *moldy beans*.

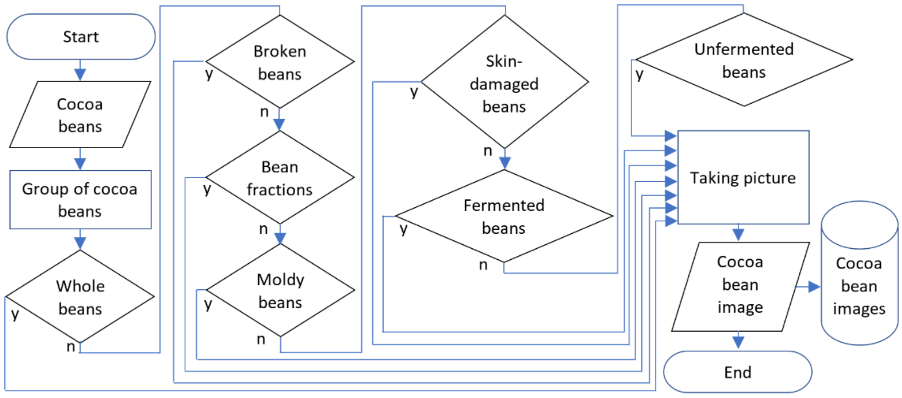


Figure 3. Flowchart of the preparation of the dataset. y—yes; n—no.

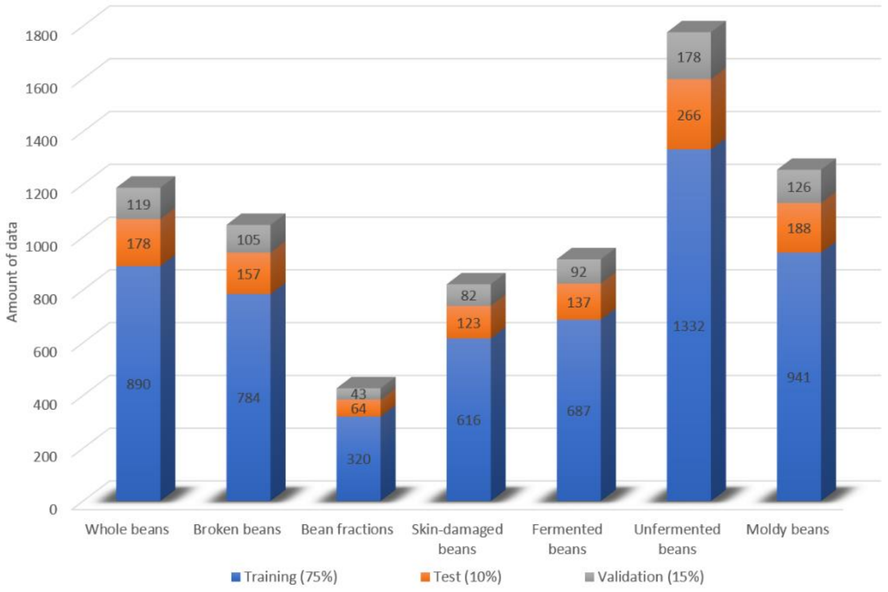


Figure 4. Class distribution of cocoa bean image datasets.

3.1.2. Methodology

The flowchart of the proposed methodology, as shown in Figure 5, consists of several steps.

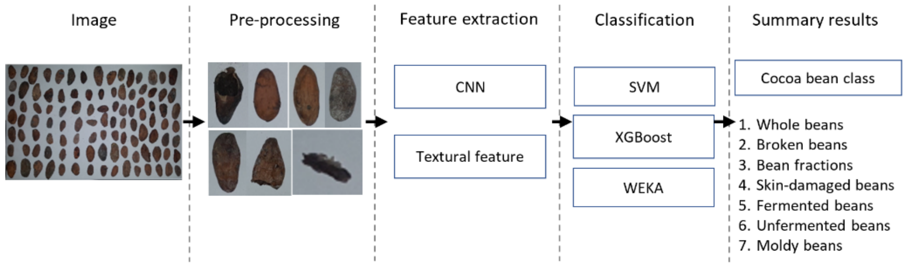


Figure 5. Proposed architecture and methodology. CNN, convolutional neural network; SVM, support vector machine; XGBoost, extreme gradient boosting; WEKA, Waikato environment for knowledge analysis.

The remainder of this paper is as follows: the first part is data retrieval; the second part is data preprocessing; the next part is feature extraction; and the last part is the classification and summary of the results.

Experimental Environment

All experiments for this study were performed on a Dual-Core Intel Core i5 laptop @ 2.7 GHz with 8 GB 1867 MHz DDR3 of RAM. The languages utilized to implement the experimental study for the proposed approach were MATLAB R2019a, Python Programming Language, and Waikato environment for knowledge analysis (WEKA) software [38].

Image Data Preprocessing

The preprocessing of digital images represents a start to ensure smoothness and success in the subsequent digital image processing steps. This method includes enhancing the image quality (contrast, brightness, etc.), noise reduction, image adjustment or reconstruction, image transformation, and determination of the part of the image to be evaluated. Recent research has shown that the image classification method might properly label images when removing background objects (i.e., only based on background content) [39,40].

By experimental evidence, visual hints can be explicitly learned independently and then combined to achieve higher performance, which validates the proposed framework's advantages. The preprocessing of cocoa bean digital images is performed by applying simple object detection (segmentation and feature extraction), measurement, and filtering. First, all of the objects are obtained, and then the results are filtered to separate targets of distinct sizes.

Feature Extraction: Convolutional Neural Network (CNN)

We used cocoa bean digital image data; in the first step, the cocoa bean images were labeled according to [35–37], saved to our local storage, and loaded into an array based on each class. This full dataset array was split into 75% for training, 10% for testing, and 15% for validation, and then a CNN was built using the architecture layer in Table 1.

Table 1. CNN architecture.

Architecture Layer	Output	Parameter Size
Convolutional 1	(222, 222, 32)	896
Convolutional 2	(220, 220, 32)	9248
Pooling	(44, 44, 32)	0
Flattening	(61,952)	0
Fully connected layer	(7)	433,671
Total parameter: 443.815		
Trainable parameter: 443.815		
Non-trainable parameter: 0		

In our CNN architecture for feature extraction, we used several CNN layers. First is the convolutional layer, which generates a feature map to predict the class probabilities for a particular feature by applying a filter that examines the complete image several pixels at a time.

The second layer is the pooling layer (down-sampling); the down-sampling of the pooling layer process consists of decreasing the image size while preserving its essential characteristics. This layer reduces the number of parameters and computations in the network, while the process in this layer enhances the performance of the network and avoids over-learning.

The next layer is the flattening layer, in which flattening the outputs generated by the previous layers are transformed into a single vector as an input for the next layer. Lastly, the final layer is the fully connected layer. In this layer, the process scales down the amount of information and maintains the essential information (the convolutional and pooling layers are usually repeated several times).

We trained our model by applying dataset augmentation using the parameters in Table 2, and using epochs of 5, 10, 15, 20, and 25, we extracted features using an intermediate model, resulting in the fully connected layer being the "output" layer of the CNN. We saved this feature extraction data to become the input for the classification process. Details of the CNN feature extraction results can be seen in Figures 6–10 and Table 3.

Table 2. Image data augmentation parameters.

Parameter	Parameter Value
Image rotation	25
Image shift width	0.1
Image shift height	0.1
Image shear	0.2
Image zoom	0.2
Horizontal flip	True
Image fill	"nearest"

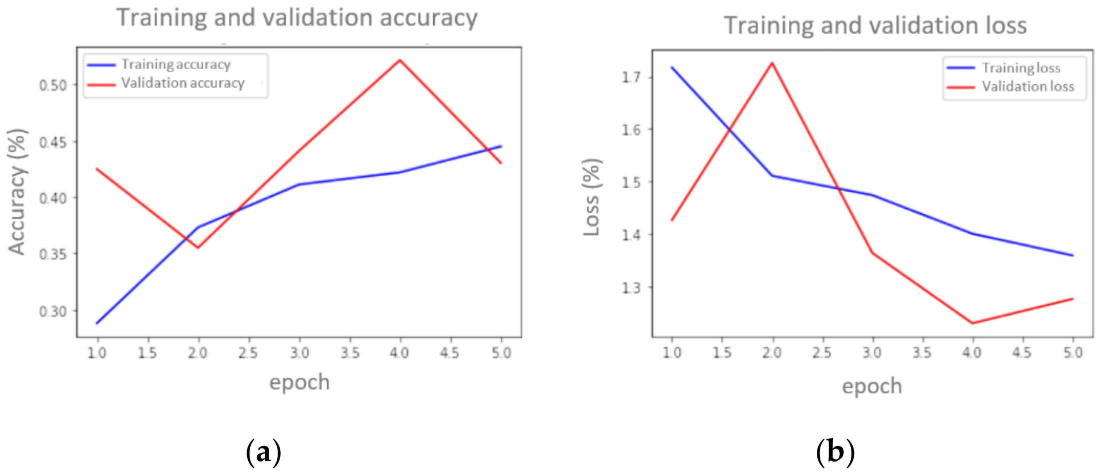


Figure 6. Validation and training: epoch = 5. (**a**) Validation training and accuracy. (**b**) Validation and training loss.

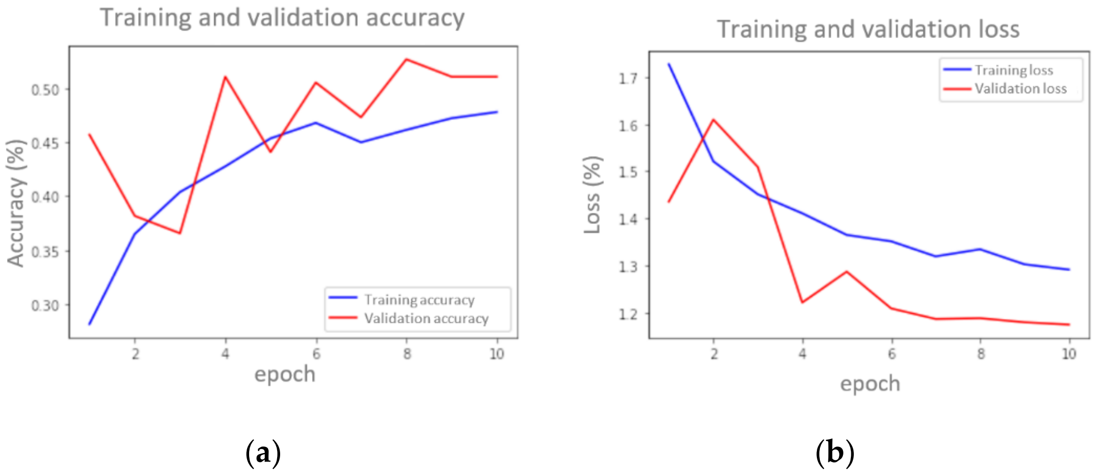


Figure 7. Validation and training: epoch = 10. (**a**) Validation and training accuracy. (**b**) Validation and training loss.

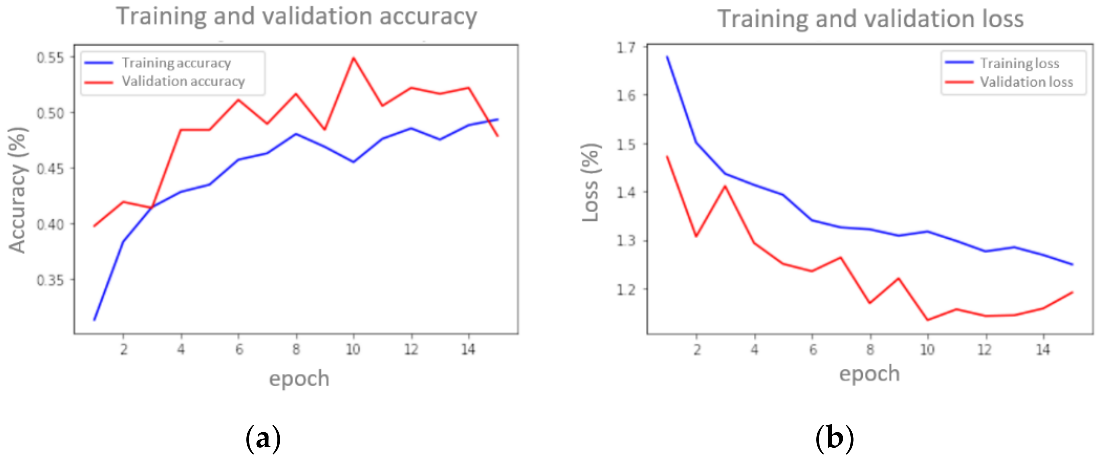


Figure 8. Validation and training: epoch = 15. (**a**) Validation and training accuracy. (**b**) Validation and training loss.

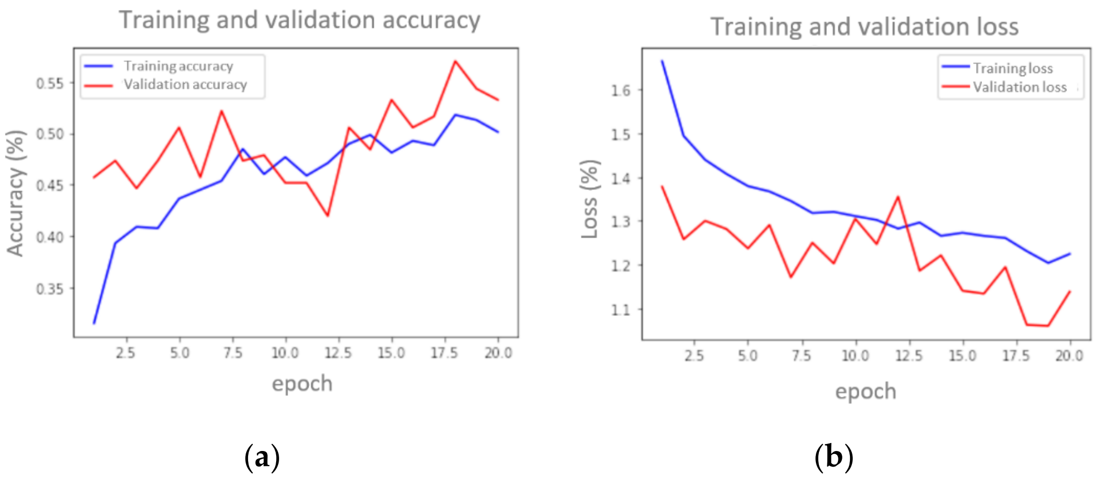


Figure 9. Validation and training: epoch = 20. (**a**) Validation and training accuracy. (**b**) Validation and training loss.

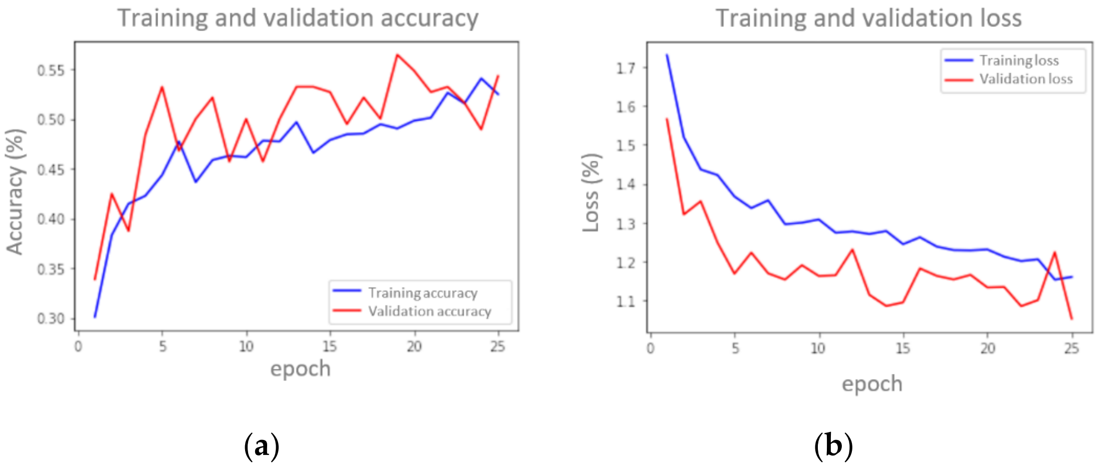


Figure 10. Validation and training: epoch = 25. (**a**) Validation and training accuracy. (**b**) Validation and training loss.

Table 3. CNN architecture model evaluation.

Model Evaluation/Epoch	5	10	15	20	25
Test loss	1.37%	1.23%	1.30%	1.28%	1.13%
Test accuracy	0.41%	0.52%	0.45%	0.47%	0.56%

Image data augmentation is a technique that implements artificial enlarging of the size of a training dataset by creating a modified version of that dataset's image. This process can improve the model's performance and ability to generalize. A detailed description of the parameters for our image data augmentation process is shown in Table 2.

An epoch is a series of steps for learning in a neural network. An example of an epoch is when the entire dataset has gone through the neural network training process once until it is returned to the beginning for the next round [41]. For our method, we used an epoch of 5, 10, 15, 20, and 25 on our feature extraction experimental parameters for the CNN feature extraction. Figures 6–10 show the accuracy and loss for the feature extraction method applied to the cocoa bean dataset. Training loss is the summation of an error made on the training dataset, and it also implies the model performance behavior after each iteration. Meanwhile, validation loss is a result error after running the validation dataset through the previously trained network.

As depicted in Table 3, as the epoch number is increased, the accuracy evaluation model increases, while the loss value decreases.

Feature Extraction: Gray Level Co-Occurrence Matrix (GLCM) Textural Features

The GLCM method is essential in the statistical approach for extracting texture features. The mathematical calculation uses a gray degree distribution (histogram) by measuring the degree of contrast, granularity, and roughness of an area of the relationship between neighbor's pixels in the image. In the first step of the GLCM calculation, the average values of the red, green, and blue pixels are taken to return the grayscale value so as to achieve reasonable gray approximation. Then, the weighted sum of the red, green, and blue (RGB) channel is used with the BT.601 standard [42], with the coefficients R = 0.30, G = 0.59, and B = 0.11.

The second step is calculating the co-occurrence matrix with a maximum value of the grayscale pixels. After calculating the co-occurrence matrix, the symmetrical matrix is created by copying a new transposed copy of the co-occurrence matrix. This copy of the matrix is added to the co-occurrence matrix, which produces a symmetrical matrix. The next step is normalizing the symmetrical GLCM by dividing each symmetrical GLCM by the sum of all elements. The final step is computing the textural features from the normalized GLCM matrix [43].

The textural features extracted from each of the GLCM features used in this research are as follows:

$$Contrast = \sum_{i,j=0}^{N_g-1} p(i,j)\left|i-j\right|^2, \tag{1}$$

$$Dissimilarity = \sum_i \sum_j \left|i-j\right| \cdot p(i,j), \tag{2}$$

$$Inverse\ Different\ Moment = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j), \tag{3}$$

$$Angular\ Second\ Moment = \sum_i \sum_j \{p(i,j)\}^2, \tag{4}$$

$$Energy = \sum_i \sum_i p(i-j)^2, \tag{5}$$

$$Correlation = \frac{\sum_i \sum_j (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}, \tag{6}$$

$$Sum\ of\ Squares\ Variance = \sum_i \sum_i (i-\mu)^2 p(i,j). \tag{7}$$

Notation:

(i, j)	GLCM coordinates, each ranging 0 to $N_g - 1$.
$p(i, j)$	(i, j)th entry in a normalized gray-tone spatial dependence matrix.
N_g	Number of distinct gray levels in the quantized image.
(x, y)	Pictorial information is represented as a function of two variables.
μ_x, μ_y	First-order statistical moments of the quantized image.
$\sigma_x \sigma_y$	Second-order statistical moments of the quantized image.

A generated GLCM feature matrix can represent a picture with fewer parameters [3,13] using the GLCM textural feature properties. When applying this method, as represented in Figure 11, we used image datasets and applied GLCM texture extraction to the dataset. For the computation of the co-occurrence matrix implemented in this study, the distance = 1 and the angles were 0°, −180°, −90°, 90°, and 180°. Table 4 provides an example of the extracted features with distance = 1 and horizontal angle = 0°.

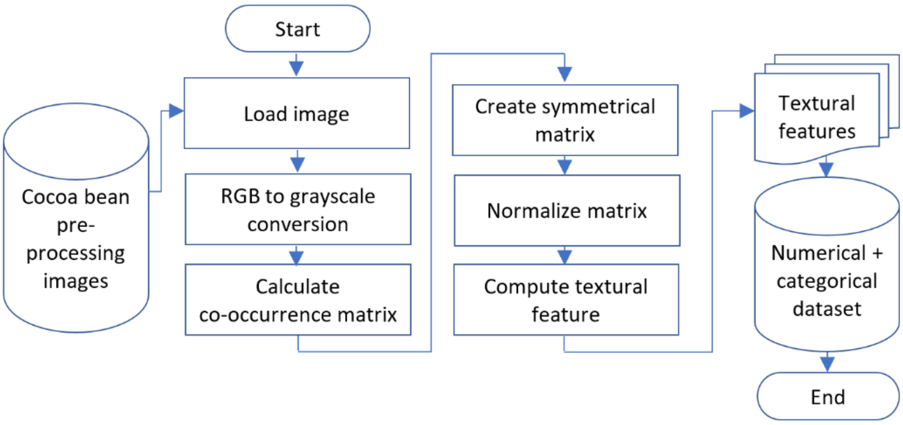

Start
Cocoa bean pre-processing images
Load image
RGB to grayscale conversion
Calculate co-occurrence matrix
Create symmetrical matrix
Normalize matrix
Compute textural feature
Textural features
Numerical + categorical dataset
End

Figure 11. Gray level co-occurrence matrix (GLCM) feature extraction. RGB—red, green, and blue.

Table 4. Features extracted using GLCM texture.

Class/Feature	1	2	3	4	5	6	7
1	28.95	1.83	0.80	0.30	0.55	0.96	369.05
2	48.56	3.17	0.58	0.22	0.47	0.95	539.75
3	22.43	1.41	0.78	0.18	0.42	0.97	491.84
4	44.61	2.07	0.75	0.15	0.38	0.97	941.58
5	26.21	1.64	0.80	0.27	0.52	0.97	415.37
6	30.87	2.45	0.56	0.16	0.39	0.94	258.46
7	36.79	2.46	0.68	0.36	0.60	0.95	401.79

Note: Class: 1—*whole beans*. 2—*broken beans*. 3—*bean fractions*. 4—*skin-damaged beans*. 5—*fermented beans*. 6—*unfermented beans*. 7—*moldy beans*. Feature: 1—contrast. 2—dissimilarity. 3—inverse different moment. 4—angular second moment. 5—energy. 6—correlation. 7—sum of squares variance.

Seven features, i.e., contrast, dissimilarity, inverse difference moment, angular second moment, energy, correlation, and sum of squares variance, were extracted, as depicted in Table 4. Seven classes of cocoa bean classifications based on SNI 2328:2008 [37], i.e., *whole beans*, *broken beans*, *bean fractions*, *skin-damaged beans*, *fermented beans*, *unfermented beans*, and *moldy beans*, were extracted from the cocoa bean digital images.

Classification Prediction

For the next step of our method, based on Figure 2, we implemented the classification process using our dataset. As previously mentioned, we split our dataset into 75% for training, 10% for testing, and 15% for validation. We used the testing (10%) and validation (15%) data and then implemented the classification process to a model that had been previously trained.

This implementation was carried out using:

1. The feature extraction model using CNN;
2. The feature extraction model using GLCM.

In addition to these two models, we also carried out a classification process using WEKA to compare the classification results. WEKA is one of the open source tools for training and testing deep learning data without programming code. We ran an experiment using the following steps: loading cocoa bean digital image data that were previously preprocessed using GLCM textural features in the attribute-relation file format (ARFF) (Figure 12), and then running the experiment based on the flow shown in Figure 13 to retrieve the classification results.

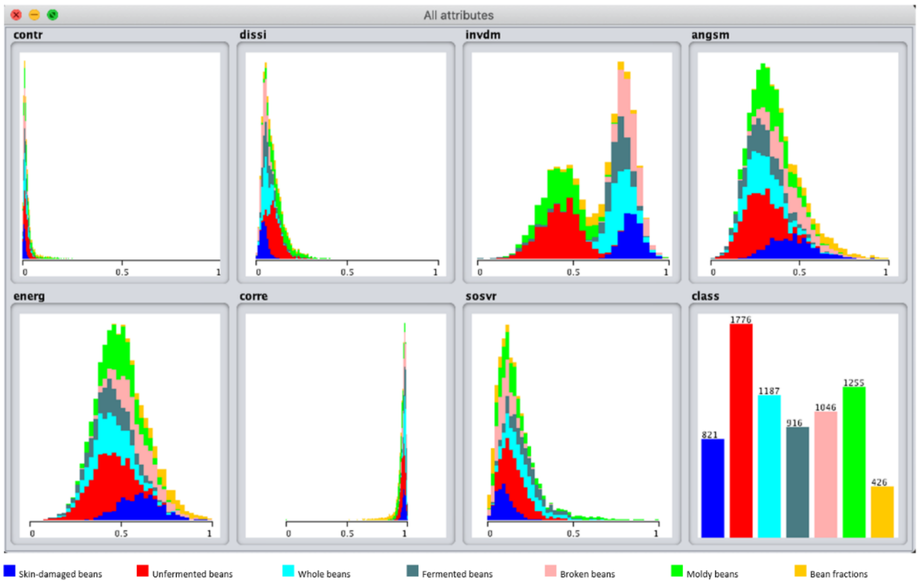


Figure 12. Data visualization. contr—contrast. dissi—dissimilarity. invdm—inverse different moment. angsm—angular second moment. energy—energy. corre—correlation. sosvr—sum of squares variance.

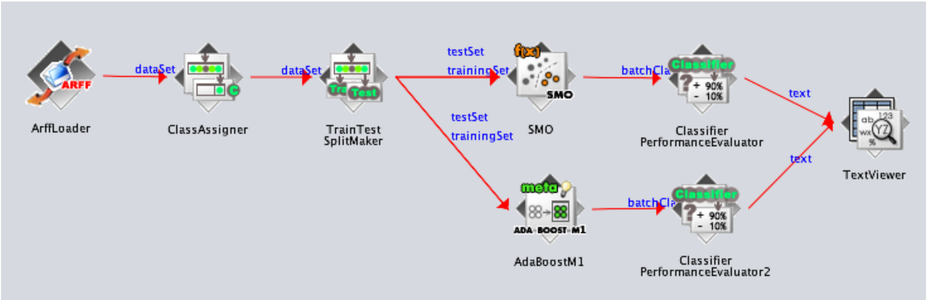


Figure 13. WEKA classification flow. Arff—attribute-relation file format. SMO—sequential minimal optimization.

In the data visualization shown in Figure 12, seven extracted GLCM feature values are displayed in graphical form, and the distribution of values for each feature is in the 0–1 value range.

In addition to these two models, we also carried out a classification process using WEKA to compare the classification results, as shown in Figure 13.

3.2. Results

To analyze the performance of the two feature extraction methods, we applied two types of classifiers—SVM (“linear” kernel) and extreme gradient boosting (XGBoost)—and compared the results to that from utilizing the WEKA tools (SVM and AdaBoost), as shown in Table 5.

From Table 5, we can summarize our research results as follows:

- A large number of epochs have an immense effect on the accuracy of the identification of the image being tested. The larger the number of epochs, the more accurate the identification of the image. As presented in Table 5 (bold formatted numbers emphasize higher accuracy values), using features extracted from the CNN method can achieve an accuracy of 59.14% with the SVM classifier and 56.99% with the XGBoost classifier.
- Based on GLCM textural features, we can see that the −90°, and 90° for the GLCM textural features method can achieve an accuracy of 61.04% with the SVM classifier and 65.08% with the XGBoost classifier.

- We also conducted experiments using WEKA in comparison with the GLCM textural features method, which achieved an accuracy of 63.86% with the SVM classifier and 39.74% with the AdaBoost classifier.

Table 5. Evaluation of results: accuracy. (Bold formatted numbers emphasize higher accuracy values).

Classifier	Epoch: 5	Epoch: 10	Epoch: 15	Epoch: 20	Epoch: 25
SVM	53.05%	54.84%	50.54%	55.91%	**59.14%**
XGBoost	51.9%	52.69%	46.95%	54.12%	**56.99%**
	GLCM (0°)	GLCM (−180°)	GLCM (−90°)	GLCM (90°)	GLCM (180°)
SVM	59.07%	59.07%	**61.04%**	**61.04%**	59.07%
XGBoost	63.02%	63.02%	**65.08%**	**65.08%**	63.02%
WEKA SVM	61.12%	61.12%	**63.86%**	**63.86%**	61.12%
WEKA AdaBoost	39.41%	39.41%	**39.74%**	**39.74%**	39.41%

Our results show that using the GLCM textural features method for feature extraction can achieve better and promising classification accuracy in comparison to CNN feature extraction. As our study was concentrated on the used/proposed methodology of feature extraction, we can conclude that our results are acceptable.

4. Discussion

Our proposed method, when utilizing CNN feature extraction, achieved an accuracy of 59.14% and 56.99% with the SVM and XGBoost classifiers, 61.04% and 65.08%, respectively, when using GLCM textural features.

The highest rate achieved by [22] with the SVM classifier resulted in a discrimination rate of 100% for the prediction and training sets. The authors in [25] proposed six CNN layers for dataset 1 and fine-tuned the pre-trained model, resulting in an accuracy of 99.49% and 99.75% for the first model in dataset 1, and 85.43% and 96.75% for the second dataset. In [26], the authors used SVM and the genetic algorithm with the features of shape, texture, and color, thereby achieving classification rates of 96–98% for an apple, 95–97% for grapes, and 96–97% for bananas. From the four-class classification conducted by [23], classification rates of 84% for *whole beans*, 52% for *broken beans*, 20% for *bean fractions*, and 20% for *skin-damaged beans* were achieved due to the non-uniform shape of the cocoa beans. The authors in [27] used expensive sensor equipment such as liquid and gas chromatography and the machine learning method for classification to investigate the fermentation rate of cocoa beans, resulting in the following misclassification rates: bootstrap forest, 9.40%; ANN, 12.80%; boosted tree, 13.60%.

It appears that compared to other farming products, the classification of cocoa beans is still a challenging problem; however, our results provide a more homogeneous classification rate for all classes.

5. Conclusions

We demonstrated our method by implementing a textural feature extraction of cocoa beans, digital images, and classifications, consisting of seven classes of beans. By utilizing the co-occurrence matrix feature of GLCM, we extracted seven features.

In our method, the CNN feature design repeatedly applies and collects all features found from several runs with a weighted sum of RGB and data augmentation with epochs of 5, 10, 15, 20, and 25. It can be used for representation purposes of the dataset features, and increasing the number of epochs can increase the final recognition rates.

From our results, we can conclude that our method, utilizing the GLCM texture features method as one part of the feature extraction process from cocoa bean digital images, can achieve promising results compared to the CNN feature extraction method. There is still the possibility for further investigation to improve the accuracy of this or other feature extraction methods in the future.

The findings from this study make several contributions to the current smart farming literature. The first is the implementation of smart farming [29] combined with IoT technology by [1,2,30], using classification methodology that employs SVM [22], CNN [25], SVM and a genetic algorithm (GA) [26], bootstrap forest, ANN, and boosted tree [27]. Next is the implementation of a low-power wireless sensor network [31,32,34] and, finally, integration with cloud computing [33].

Author Contributions: Conceptualization, S.W.P., M.K., and J.-S.L.; data curation, Y.A.; formal analysis, S.W.P.; investigation, Y.A.; resources, Y.A.; supervision, M.K. and J.-S.L.; visualization, S.W.P.; writing—original draft, Y.A.; writing—review and editing, S.W.P. All authors read and agreed to the published version of the manuscript.

Funding: This study was supported by a collaborative research project between the Kyushu Institute of Technology (Kyutech) and the National Taiwan University of Science and Technology (Taiwan-Tech).

Acknowledgments: The authors gratefully acknowledge the support by the Kyushu Institute of Technology—National Taiwan University of Science and Technology Joint Research Program, under Grant Kyutech-NTUST-107-06.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Farooq, M.S.; Riaz, S.; Abid, A.; Abid, K.; Naeem, M.A. A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *IEEE Access* **2019**, *7*, 156237–156271. [CrossRef]
2. Yoon, C.; Huh, M.; Kang, S.; Park, J.; Lee, C. Implement Smart Farm with IoT Technology. In Proceedings of the 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon-si Gangwon-do, Korea, 11–14 February 2018; pp. 749–752.
3. Haralick, R.M.; Shanmugam, K.; Dinstein, I. Textural Features for Image Classification. *IEEE Trans. Syst. Man Cybern.* **1973**, *SMC-3*, 610–621. [CrossRef]
4. Ye, F.; Shi, Z.; Shi, Z. A comparative study of PCA, LDA and Kernel LDA for image classification. In Proceedings of the International Symposium on Ubiquitous Virtual Reality, Gwangju, South Korea, 8–11 July 2009; pp. 51–54.
5. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
6. Purwanto, D.; Chen, Y.; Fang, W. First-Person Action Recognition with Temporal Pooling and Hilbert-Huang Transform. *IEEE Trans. Multimed.* **2019**, *21–12*, 3122–3135. [CrossRef]
7. Nazir, F.; Majeed, M.N.; Ghazanfar, M.A.; Maqsood, M. Mispronunciation Detection Using Deep Convolutional Neural Network Features and Transfer Learning—Based Model for Arabic Phonemes. *IEEE Access* **2019**, *7*, 52589. [CrossRef]
8. Khagi, B.; Lee, C.G.; Kwon, G. Alzheimer's disease Classification from Brain MRI based on transfer learning from CNN. In Proceedings of the 11th Biomedical Engineering International Conference (BMEiCON), Chiang Mai, Thailand, 21–24 November 2018; pp. 1–4.
9. Sengur, A.; Akhtar, Z.; Akbulut, Y.; Ekici, S.; Budak, U. Deep Feature Extraction for Face Liveness Detection. In Proceedings of the International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 September 2018; pp. 1–4.
10. Zhang, Y.; Gao, X.; He, L.; Lu, W.; He, R. Objective Video Quality Assessment Combining Transfer Learning with CNN. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 2716–2730. [CrossRef]
11. Löfstedt, T.; Brynolfsson, P.; Asklund, T.; Nyholm, T.; Garpebring, A. Gray-level invariant Haralick texture features. *PLoS ONE* **2019**. [CrossRef]
12. Öztürk, S.; Akdemir, B. Application of Feature extraction and Classification Methods for Histopathological Image using GLCM, LBP, LBGLCM, GLRLM and SFTA. In Proceedings of the Part of special issue: International Conference on Computational Intelligence and Data Science, Gurgaon, India, 7–8 April 2018; Volume 132, pp. 40–46.
13. Brynolfsson, P.; Nilsson, D.; Torheim, T.; Asklund, T.; Karlsson, C.T.; Trygg, J.; Nyholm, T.; Garpebring, A. Haralick texture features from apparent diffusion coefficient (ADC) MRI images depend on imaging and pre-processing parameters. *Sci. Rep.* **2017**, *7*, 4041. [CrossRef]

14. Soh, L.K.; Tsatsoulis, C. Texture Analysis of SAR Sea Ice Imagery Using Gray Level Co-Occurrence Matrices. *IEEE Trans. Geosci. Remote Sens.* **1999**, *37*, 780–795. [CrossRef]
15. Clausi, D.A. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Can. J. Remote Sens.* **2002**, *28*, 45–62. [CrossRef]
16. Jachowski, N.R.A.; Quak, M.S.Y.; Friess, D.A.; Duangnamon, D.; Webb, E.L.; Ziegler, A.D. Mangrove biomass estimation in Southwest Thailand using machine learning. In *Applied Geography*; Elsevier: Amsterdam, The Netherlands, 2013; Volume 45, pp. 311–321.
17. Kamilaris, A.; Prenafeta-Boldu', F.X. *Deep learning in agriculture: A survey. Computers and Electronics in Agriculture*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 147, pp. 70–90.
18. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine Learning in Agriculture: A Review. *Sensors* **2018**, *18*, 2675. [CrossRef]
19. Kumar, R.; Singh, M.P.; Kumar, P.; Singh, J.P. Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique. In Proceedings of the International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Chennai, India, 6–8 May 2015; pp. 138–145.
20. Behera, S.K.; Rath, A.K.; Mahapatra, A.; Sethy, P.K. Identification, classification & grading of fruits using machine learning & computer intelligence: A review. *J. Ambient Intell. Human. Comput.* **2020**. [CrossRef]
21. Zawbaa, H.M.; Hazman, M.; Abbass, M.; Hassanien, A.E. Automatic fruit classification using random forest algorithm. In Proceedings of the IEEE International Conference on Hybrid Intelligent Systems, Kuwait, Kuwait, 14–16 December 2014; pp. 164–168.
22. Yro, A.; Camille, E.N.; Kpalma, K. Cocoa Beans Fermentation Degree Assessment for Quality Control Using Machine Vision and Multiclass SVM Classifier. *Int. J. Innov. Appl. Stud. ISSR J.* **2018**, *24*, 1711–1717.
23. Astika, I.W.; Solahudin, M.; Kurniawan, A.; Wulandari, Y. Determination of Cocoa Bean Quality with Image Processing and Artificial Neural Network. In Proceedings of the AFITA 2010 International Conference, The Quality Information for Competitive Agricultural Based Production System and Commerce, Bogor, Indonesia, 3–7 October 2010.
24. Unser, M. Sum and Difference Histograms for Texture Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 118–125. [CrossRef] [PubMed]
25. Hossain, M.S.; Al-Hammadi, M.; Muhammad, G. Automatic Fruits Classification Using Deep Learning for Industrial Applications. *IEEE Trans. Ind. Inform.* **2018**, *15*, 1027–1034. [CrossRef]
26. Mahajan, M.S.; Patil, S.T. Optimization and Classification of Fruit using Machine Learning Algorithm. *IJIRST Int. J. Innov. Res. Sci. Technol.* **2016**, *3*, 377–381.
27. Tan, J.; Balasubramanian, B.; Sukha, D.A.; Ramkissoon, S.; Umaharan, P. Sensing fermentation degree of cocoa (*Theobroma cacao* L.) beans by machine learning classification models based electronic nose system. *J. Food Process Eng.* **2019**, 42. [CrossRef]
28. Knapp, A.W. *Cocoa and Chocolate: Their History from Plantation to Consumer*; Chapman and Hall Ltd.: London, UK, 1920; pp. 36–40.
29. Bacco, M.; Barsocchi, P.; Ferro, E.; Gotta, A.; Ruggeri, M. The Digitization of Agriculture: A Survey of Research Activities on Smart Farming. *Array* **2019**, *3–4*, 1–11.
30. Wang, P.; Valerdi, R.; Zhou, S.; Li, L. Introduction: Advances in IoT research and applications. *Springer Inform. Syst. Front.* **2015**, *17*, 239–241. [CrossRef]
31. Baldovino, R.G.; Valenzuela, I.C.; Dadios, E.P. Implementation of a Low-power Wireless Sensor Network for Smart Farm Applications. In Proceedings of the IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, 29 November–2 December 2018; IEEE: New York, NY, USA, 2018; pp. 1–5.
32. Codeluppi, G.; Cilfone, A.; Davoli, L.; Ferrari, G. LoRaFarM: A LoRaWAN-Based Smart Farming Modular IoT Architecture. *Sensors* **2020**, *20*, 2028. [CrossRef]
33. Botta, A.; Donato, W.; Persico, V.; Pescapé, A. Integration of cloud computing and Internet of Things: A survey. *Future Gener. Comput. Syst.* **2016**, *56*, 684–700. [CrossRef]
34. Navulur, S.; Sastry, A.S.C.S.; Prasad, M.N.G. Agricultural management through wireless sensors and Internet of Things. *Int. J. Electr. Comput. Eng.* **2017**, *7*, 3492–3499. [CrossRef]

35. Badan Standardisasi Nasional (BSN). *Petunjuk Pengambilan Contoh Padatan SNI-19-0428-1998*; Badan Standardisasi Nasional: Jakarta, Indonesia, 1998.
36. Kementrian Pertanian. *Peraturan Menteri Pertanian Republik Indonesia: Pedoman Penanganan Pascapanen Kakao Nomor. 51/Permentan/OT.140/9/2012.*; Kementrian Pertanian: Jakarta, Indonesia, 2012.
37. Badan Standardisasi Nasional (BSN). *Biji kakao SNI 2323:2008 ICS 1.67.140.30 Kakao.*; Badan Standardisasi Nasional: Jakarta, Indonesia, 2008.
38. Frank, E.; Hall, M.A.; Witten, I.H. *The WEKA Workbench, Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed.; Morgan Kaufmann: Burlington, MA, USA, 2016.
39. Alcorn, M.A.; Li, Q.; Gong, Z.; Wang, C.; Mai, L.; Ku, W.; Nguyen, A. Strike (with) a Pose: Neural Networks Are Easily Fooled by Strange Poses of Familiar Objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4845–4854.
40. Zhu, Z.; Xie, L.; Yuille, A.L. Object recognition with and without objects. *arXiv* **2016**, arXiv:1611.06596.
41. Yani, M.; Irawan, B.; Setiningsih, C. Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry’s Nail. In Proceeding of the International Conference on Electronics Representation and Algorithm (ICERA), Yogyakarta, Indonesia, 29–30 January 2019.
42. International Telecommunication Union. *Recommendation ITU-T BT.601-7 Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-screen 16:9 Aspect Ratios BT Series Broadcasting Service*; International Telecommunication Union: Geneva, Switzerland, 2011.
43. Hall-Byer, M. *GLCM Texture: A Tutorial v. 2.7 April 2004*; Department of Geography, University of Calgary: Calgary, AB, Canada, 2004.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.