IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 September 2023, accepted 14 November 2023, date of publication 28 November 2023,
date of current version 19 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337424

RESEARCH ARTICLE

Fast and Compact Model for Palm Bunch Classification Using DT-CWT and LSTM on Hue Histogram

PRONTHEP PIPITSUNTHONSAN[1], THANATE KHAORAPAPONG[1], SUTKHET NAKASATHIEN[2], PETCHARAT SURIYACHAI[1], SITTIPORN CHANNUMSIN[3], AND MITCHAI CHONGCHEAWCHAMNAN[1], (Senior Member, IEEE)

[1]Faculty of Engineering, Prince of Songkla University, Songkhla 90110, Thailand
[2]Faculty of Agriculture, Kasetsart University, Bangkok 10900, Thailand
[3]Space Technology Research Center, Geo-Informatics and Space Technology Development Agency (GISTDA), Bangkok, Chonburi 20230, Thailand

Corresponding author: Mitchai Chongcheawchamnan (mitchai.c@psu.ac.th)

This work was supported in part by the Royal Golden Jubilee Ph.D.-Thailand (RGJPHD) under Grant Ph.D./0177/2561; in part by the Agriculture Food and Biotechnology Project of in subproject ''In-field application of oil palm bunch inspector system'' Kasetsart University (KU), Thailand, under Grant TH 4.0 (2 (2.1) 60); and in part by the Prince of Songkla University (PSU), Thailand, under Grant ENG610416M.

ABSTRACT Traditionally, the grading process of fresh palm bunches has used human observation at palm trading sites, which is not precise and unreliable due to human bias. Several studies have presented automatic grading methods based on image processing and machine learning. Unfortunately, these models are not sufficiently compact to be implemented on trading sites. Therefore, a compact grading model for automatic grading suitable for implementation at palm trading sites is presented. Two key elements make the model more compact and efficient. The first element is a reduction in the size of the input dataset. We achieved this by replacing the multi-dimensional RGB palm bunch image with a one-dimensional hue histogram. The second element was the core engine of the proposed classifier. It consists of a dual-tree complex wavelet transform (DT-CWT) connected to the LSTM back end as the front end. The proposed model was proven using real image datasets of 800 palm bunches of several varieties collected from trading sites. The robustness of the model was investigated by verifying its accuracy in several noisy environments. Based on the testing process, the proposed model achieves 91.67% accuracy at 6 dB signal-to-noise ratio (SNR).

INDEX TERMS Feature extraction, deep learning, deep neural networks, modeling.

I. INTRODUCTION

Oil palm has been a key global economic crop plants, and over 100 million farmers have cropped palm for years. Various kinds of palm oil products are available, highlighting this plant's significant worldwide importance [1]. Hence, the growing demand for oil palm products is essential for developing techniques to improve the productivity of oil palm farmers to produce fresh palm bunches (FPBs). Furthermore, a procedure or common agreement between buyers and sellers for the quality estimation and pricing of an FPB sample is needed at a trading site. Local experts have carried out such an estimation of FPB ripeness, which is relatively effective but time-consuming.

To overcome the limitations of the ripeness estimation approach, this study aims to develop a classification method for oil palm bunches. A set of FPB images was collected, pre-processed, augmented, and used to create a model for classifying the four ripeness levels of FPBs: overripe, ripe, underripe, and unripe. Ripeness depends on various factors and has been a challenging research problem, as explained next.

Color characteristics are of considerable importance when used to detect the ripeness of FPBs [2]. For instance, raw

The associate editor coordinating the review of this manuscript and approving it for publication was Mark Kok Yew Ng.

oil palm fruits display a dark purple color, which gradually changes to orange as color they reach their ripeness peak [3]. Moreover, color indicates the amount of free fatty acids in palm oil fruits, which significantly impacts the overall quality of crude palm oil. Identifying the correct stage or level of ripeness during harvesting is important for determining the composition of free fatty acids [7].

Based on expert opinions, good-quality FPBs are assessed by their color, texture, shape, and other factors [1], [2], [3], [4], [5], [6]. However, color is of paramount importance for identifying the ripeness of FPBs. In some cases, the skin color of unripe fruits may differ significantly from that of ripe fruits. Furthermore, many color spaces need to be considered when assessing the ripeness of FPBs. For example, the ripeness of oil palm fruits is classified using a method involving feature extraction in an RGB (red, green, and blue) color model [2]. This RGB model can also be transformed into other color models, such as Hue Saturation and Value (HSV), CIELAB, CMYK (Cyan, Magenta, Yellow, and Black) and HIS (Hue, Saturation, and Intensity), which are used to detect the ripeness of FPBs [4], [5], [6]. Similarly, [8] applied a combination of three models, RGB, HSI, and L*a*b, to classify FPBs. In [5] and [9], a machine was developed to classify FPBs, but the implementation of necessarily closed cabinets is required.

As described previously, this study presents a model for classifying the ripeness levels of FPB images. As part of the model construction, the issue of noise in the images must be considered and discussed.

The noise level in realistic surroundings has various effects on the quality of the implemented color model. Wavelet transform has been extensively utilized across various disciplines to effectively decompose noise signals and extract nonstationary features [10]. Moreover, the performance of the wavelet-based transform method for one, two, and three dimensions is better than that of the competing techniques [11]. The dual-tree complex wavelet transforms (DT-CWT) have been proposed for image denoising, fault detection, medical science, and electroencephalograph (EEG) signal recognition [12], [13], [14], [15].

Image denoising has been a widely recognized and active topic in the field of image processing. The primary objective is to preserve essential image details while decreasing the effect of noise. This image denoising can then reconstruct the original image from its corrupted version. The work in [16] presents three image denoising techniques: DT-CWT, Stationary Wavelet Transform, and Discrete Wavelet Transform. The experimental results in [16] show that the DT-CWT technique could restore images with a higher peak signal-to-noise ratio (SNR) and lower Mean Square Error (MSE) than the other two methods. The system consists of two parallel Discrete Wavelet Transforms utilizing isolated low-pass and high-pass filters [17], called real-tree and imaginary-tree, respectively. Additionally, it generates an analytic signal based on the fundamentals of Fourier transforms. The signal illustrates a smooth, non-oscillating magnitude, nearly invariant based on phases, reduced aliasing effects, and directional wavelets in higher dimensions.

The primary stages are necessary for ordinary noise classification and consist of data collection, data pre-processing, and model classification. Additive white Gaussian noise (AWGN) is a widely implemented technique in data pre-processing for transforming the hue channel into RGB images. Nevertheless, the implementation of this FPB grading application using RGB images in real-world scenarios may have an impact in terms of processing time and grading accuracy. From the signal theory aspect, the hue channel is a one-dimensional (1-D) signal; hence, the 1-D long short-term memory (LSTM) proposed grading of FPBs from environmental noise. On the other hand, traditional signal classification problems are mainly divided into feature extraction and pattern classification. Support vector machines (SVM), random forests (RF), nearest neighbors (NN), recurrent neural networks (RNN), long short-term memory, and convolutional neural networks are used in the feature extraction of one-dimensional signals [18], [19], [20]. In the experiment, the general grading of FPBs at real trading sites was approximately four. The hue data was transformed from the RGB image, where the four trading sites were collected. The RGB image transforming to a one-dimensional hue and mixing AWGN will lead to the overfitting of model learning. The simulated noise was mixed using signal-to-noise ratio (SNR). A fast and accurate FPB classification method is required to identify trading site operations effectively. An essential component of a real-time application is implementing large datasets and learning millions of parameters that affect the impact of high latency [21], [22], [23].

In this paper, we propose a DT-CWT combined with LSTM for grading oil palm bunches. The proposed pre-processing method based on wavelet transform can reduce noise and extract features from a one-dimensional hue channel. To improve the detection accuracy and time of the model, the main contributions of this paper are as follows:

- DT-CWT was proposed for pre-processing. The DT-CWT decomposes the required signal and noise, causing the key feature to appear prominently.
- An LSTM model is implemented to obtain a high accuracy and fast FPB grading technique.

II. METHODOLOGY

This section describes the methodology used in the proposed FPB classification. Fig. 1 shows the sequence of methods used to construct the classification model, starting from data collection, pre-processing and augmentation, deep learning model development, and model evaluation. Details of each method are provided below.

A. DATA COLLECTION

Data used in this research were collected using mobile equipment called "*PalmSnap*" [24] at four trading sites in the

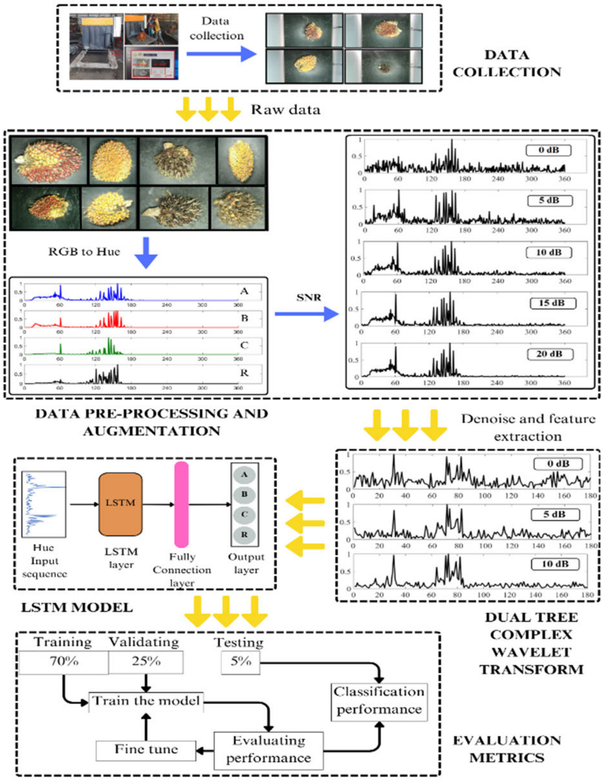


FIGURE 1. Flow diagram of the proposed method.

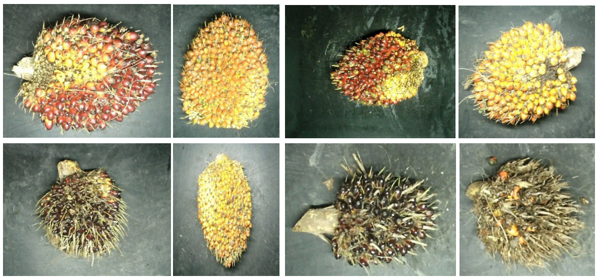


FIGURE 2. Examples of FPB image datasets.

South of Thailand for three months, from August to October. The RGB (Red, Green, Blue) images of the FPBs were recorded in a bmp format with a size of 1,920 pixels × 1,080 pixels. Fig. 2 illustrates the examples of the collected data. Next, local experts classified these images into four categories according to a ripeness level criterion, commonly used in the oil palm industry. These categories were A, B, C, and R, corresponding to the ripeness levels of overripe, ripe, underripe, and unripe, respectively. The FPB in category R is generally undesirable and likely to be rejected at most trading sites.

B. PRE-PROCESSING AND AUGMENTATION

After the data collection described in the previous subsection, the RGB image files were preprocessed and augmented before classification. Specifically, the color model in these files was converted to the Hue, Saturation, Value (HSV) color model, and only the hue component of this model was used. The Hue is an angle from 0° to 360°, representing the three primary colors (red, blue, and yellow) and the three secondary colors (orange, green, and violet) in the color wheel. Equation (1) shows the numerical representation of the primary colors or H used for the three primary colors in this study. The Hue channel is a one-dimensional signal with an 8-bit data point.

$$H = \begin{cases} 60^{\circ} \times \left(\dfrac{G' - B'}{C_{max} - C_{min}}\right) mod\ 6, & C_{max} = R' \\ 60^{\circ} \times \left(\dfrac{B' - R'}{C_{max} - C_{min}}\right) + 2, & C_{max} = G' \\ 60^{\circ} \times \left(\dfrac{R' - G'}{C_{max} - C_{min}}\right) + 4, & C_{max} = B' \end{cases} \quad (1)$$

Hue is the fundamental attribute that illustrates the color the human visual system recognizes. The measured span of the hue channel was 360°, with red at 0°, green at 120°, and blue at 240°. The hue color space can be modified to include a wider spectrum of colors such as orange, yellow, cyan, magenta, pink, and brown. The hue data consisted of a one-dimensional signal with 360 data points. The signal data will be different owing to the color space in the level ripeness of the FPBs when transforming to a hue channel.

At trading sites, factors such as camera quality and surrounding light can impair the quality of images taken by mobile equipment. Thus, images may contain 'noise' interfering with the FPB classification model. AWGN was added to the pre-processed data in our experiment to simulate a similar environment. The SNR in decibels is defined in (2), where P_s is the power of the input signal, and P_n is the power of the input noise.

$$SNR = 10log_{10}\frac{P_s}{P_n} \quad (2)$$

Data augmentation is commonly performed when the amount of data in each category in the dataset is limited imbalanced. Augmentation aims to avoid an overfitting problem during the learning phase of deep learning. In this study, the synthetic minority over-sampling technique (SMOTE) was deployed for data augmentation to address this problem. Furthermore, the dataset of all grades is 800 images, with each grade comprising 200 images. All images were also subjected to data pre-processing, as described above.

C. DUAL TREE COMPLEX WAVELET TRANSFORM

Based on the Fourier transform, DT-CWT has gained much interest for denoising and extracting features from a one-dimensional hue. The DT-CWT incorporates a directional analysis component suitable for nonstationary signals demonstrating directional characteristics or features dependent on orientation. Moreover, the DT-CWT can extract important features from nonstationary signals, which can be useful for numerous applications, such as image processing, audio analysis, and biomedical signal processing,

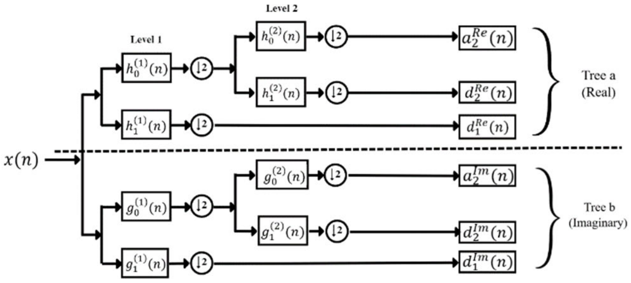


FIGURE 3. Two-level DT-CWT decomposition operation [30].

including electroencephalography and electrocardiography signals [26], [27]. DT-CWT was also used to extract features from electroencephalography (EEG) and electrocardiography signals for classification applications [28], [29]. It is generally known that the window size for the DT-CWT should be chosen appropriately. Although applying FFT to time-varying nonstationary signals allows the extraction of one-dimensional components, it does not provide statistical information on the temporal development of these components. Therefore, a one-dimensional DT-CWT was proposed for denoising and feature extraction.

The basic structure of the DT-CWT shown in Fig. 3 consists of two isolated trees, namely trees a and b, which represent real and imaginary components. DT-CWT decomposes an input real signal into complex wavelet coefficients using real and imaginary trees. Each tree comprises of several filter banks depending on the number of levels. The diagram in Fig. 3 shows the structure of a one-dimensional DT-CWT with two levels. The low-pass filter coefficients of level i for the real and imaginary components are denoted by $h_0^{(i)}$ and $g_0^{(i)}$, respectively, whereas the high-pass filter coefficients of level i for the real and imaginary components are denoted by $h_1^{(i)}$ and $g_1^{(i)}$.

The wavelet function of one-dimensional DT-CWT is shown in (3) and (4). Two real wavelet functions were used as a plural form expression's real and imaginary parts. The Hilbert transform of the real part approximates the imaginary parts.

$$\varphi\left(t\right)=\varphi_h\left(t\right)+j\varphi_g\left(t\right) \tag{3}$$

$$\varphi_g\left(t\right)=H\left\{\varphi_h\left(t\right)\right\}. \tag{4}$$

where $H\{.\}$ where is the Hilbert transform operator. The algorithm analyzes the input signal and synthesizes a series of complex wavelet coefficients that characterize both the real and imaginary components of the signal. The wavelet and scale coefficients of tree a are given by following:

$$d_j^{Re}\left(n\right)=2^{\frac{j}{2}}\int_{-\infty}^{\infty}x\left(t\right)\varphi_h(2^jt-n)dt,\quad (j=1,2,..J) \tag{5}$$

$$a_J^{Re}\left(n\right)=2^{j/2}\int_{-\infty}^{\infty}x\left(t\right)\varphi_h(2^Jt-n)dt \tag{6}$$

where J is the total number of decomposition layers, and the wavelet and scale coefficients of the tree b are:

$$d_j^{Im}\left(n\right)=2^{\frac{j}{2}}\int_{-\infty}^{\infty}x\left(t\right)\varphi_g(2^jt-n)dt,\quad (j=1,2,..J) \tag{7}$$

$$d_J^{Im}\left(n\right)=2^{j/2}\int_{-\infty}^{\infty}x\left(t\right)\varphi_g(2^Jt-n)dt \tag{8}$$

D. LSTM MODEL

LSTM represents an important development over recurrent neural networks (RNNs), originally designed for predicting sequence activities, owing to its effective reduction of the stability limitation, commonly referred to simply as the vanishing gradient problem. The gating function with state dynamics in LSTM was improved for nonlinearity [31]. The LSTM network comprises multiple memory blocks that connect through layers. Each layer includes multiple sets of memory cells that communicate recurrently. In addition to three multiplicative units known as the input, forget, and output gates [21], input and output gate functions are employed to regulate the transmission of input and output signals from the memory cell to the remaining components of the network. Furthermore, a forget gate is incorporated into the memory cell, facilitating the transmission of output information with significant weights from the preceding neuron to the subsequent neuron. Retaining information in memory is contingent upon the outcome of high activation; specifically, if the input unit exhibits high activation, the information is stored in a memory cell. When the activation level of the output unit is high, it transmits information to the subsequent neuron. Alternatively, data with significant weights are stored in the memory cells.

The LSTM network computes the mapping between the input sequence $X=\{X\left(1\right),X\left(2\right),\ldots,X\left(n\right)\}$, where $X(t)$ and the output sequence $h=\{h\left(1\right),h\left(2\right),\ldots,h\left(n\right)\}$. The other parameters are given by:

$$i_{gate}=sigmoid(W_iX\left(t\right)+U_ih\left(t-1\right)+b_i) \tag{9}$$

$$f_{gate}=sigmoid(W_fX\left(t\right)+U_fh\left(t-1\right)+b_f) \tag{10}$$

$$o_{gate}=sigmoid(W_oX\left(t\right)+U_oh\left(t-1\right)+b_o) \tag{11}$$

$$\widetilde{C_t}=\tanh\left(W_cX\left(t\right)+U_ch\left(t-1\right)+b_c\right) \tag{12}$$

$$C_t=f_{gate}\cdot C\left(t-1\right)+i_{gate}\cdot\tilde{C_t} \tag{13}$$

$$h_t=o_{gate}\cdot\tanh C_t \tag{14}$$

W_i, W_f, W_o, W_c, U_i, U_f, U_oU_c and b_i, b_f, b_ob_c represent the weight associated with the hidden state, the input and bias variables of the three gates, and a memory cell, respectively. where $h(t)$ and $C\left(t\right)$ are the hidden and cell states, respectively. After processing (13), $C\left(t\right)$ is transformed into the current memory cell unit. Equation (14) shows the element-wise multiplication of the prior hidden unit outputs and the previous memory cell units. This is the nonlinearity on top of the three gates in the form of tanh and sigmoid activation functions, shown in (9) to (14). Here, $t-1$ and t are the previous and current time steps, respectively.

The LSTM model comprises an input layer, LSTM layer, batch normalization layer with activation function, global max pooling layer, dropout layer, fully connected layer, and an output layer with SoftMax. The one-dimensional hue's input dimension (180,1) entered the model, input to the LSTM layer, and implemented the input parameter with a hidden size of 500 units. After batch normalization with the *Relu* function and global maximum polling, the offset of the hidden unit was reduced, thereby improving the accuracy of the training. Global max-polling was integrated through a convex combination, wherein a single weighting parameter was utilized. This parameter can either be randomly selected or trained based on local features, as indicated in the network [33], [34]. The fully connected network expands all the data and maps the learned ''distributed features representation'' to the labeled sample space through the multilayer full connection layer with dropout. Finally, the classification results are presented using SoftMax.

E. EVALUATION METRICS

In the pretraining, we used data augmentation to increase the SNR set in the one-dimensional hue. The datasets were split into three groups: training, validation, and testing, which used a shuffled dataset at every epoch to avoid bias. The proportions of the three groups were 70%, 25%, and 5% for training, validation, and testing, respectively. After pretraining, the performance of the model was evaluated using seven indicators. The indicators used in this research included accuracy, precision, recall, F1-score, area under the curve (AUC), and total classification cost (TCC). Moreover, the testing procedure was used to evaluate the statistical significance of the performance differences among the models in different situations.

Various evaluation metrics such as accuracy, precision, recall, and F1-score were investigated to evaluate the classification framework's effectiveness. The metrics of accuracy, precision, and recall were derived from the total number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The performance of the model was evaluated using six metrics: accuracy (Acc.), precision (Pre.) Sensitivity (Sen.), and specificity (Spec.), F1-score and the area under the ROC curve (AUC). These metrics were computed using TP, FP, TN, and FN [35], [36]. The metrics are defined as follows:

$$Acc. = \frac{TP + TN}{TP + FP + TN + FN} \tag{15}$$

$$Pre. = \frac{TP}{TP + FP} \tag{16}$$

$$Sen. = \frac{TP}{TP + FN} \tag{17}$$

$$Spec. = \frac{TN}{FP + TN} \tag{18}$$

$$F1score = \frac{2 \times pre. \times Sen.}{Pre. + Sen.} \tag{19}$$

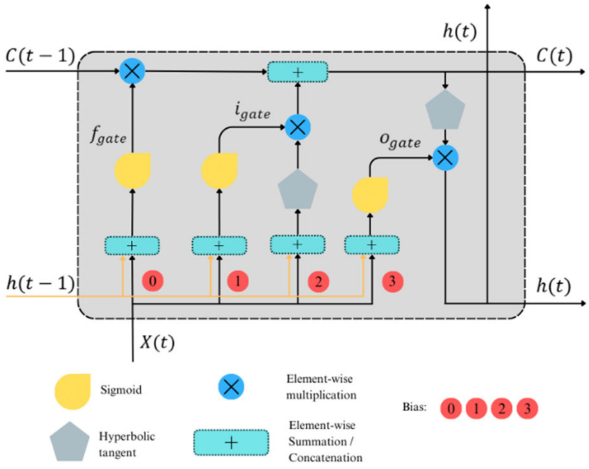


FIGURE 4. Architecture of LSTM [32].

An ROC curve, referred to as a receiver operating characteristic curve, is a visualization that illustrates the performance of a classification model across various classification thresholds. The plotted curve represents the relationship between two parameters, specifically the true positive rate (TPR) and false positive rate (FPR). TPR, commonly referred to as recall, is typically defined as follows:

$$TPR = \frac{TP}{TP + FN} \tag{20}$$

where FPR is defined as follows:

$$FPR = \frac{FP}{FP + TN} \tag{21}$$

AUC, which has recently gained significant popularity, additionally encountered challenges because of the possible utilization of various misclassification cost distributions across different classifiers [37]. Moreover, in application-oriented research, the model is challenging to implement in real-time applications because of the model size or testing time of resource overload. The Training Cost Calculator (TCC) is an excellent productivity-enhancing tool for machine learning projects. In this field of research, the parameters of the model are limited to the objective of obtaining the optimal sample processing speed and the highest level of accuracy.

III. EXPERIMENTAL RESULTS

In this section, the experimental results for the classification model are presented. First, one-dimensional hue datasets were demonstrated with different SNRs, as shown in Fig. 5. Image data transformation from RGB to HSV space was performed initially. This was performed using the ''*rgb2hsv*'' MATLAB function. One-dimensional hue data were obtained with the x-axis referring to the color degree and the y-axis indicating the corresponding intensity levels. A normalization technique was applied to reduce the data along the y-axis. This technique scales intensity to a uniform range, thereby improving the accuracy and robustness of the model. The

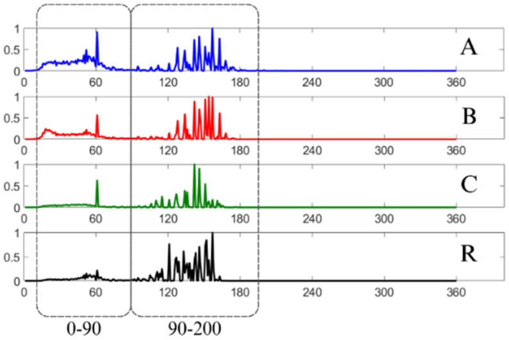


FIGURE 5. The FFBs 2D image transforms to a one-dimensional hue.

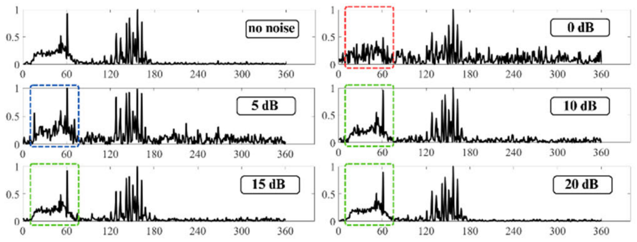


FIGURE 6. The one-dimensional hue with different SNRs.

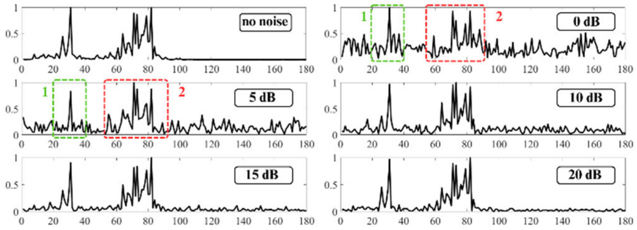


FIGURE 7. DT-CWT outputs.

TABLE 1. Comparison of six model performances.

	Parameter number (million)	Layer	Training time (s)	Model Sizes (MB)	Testing times (s)
AlexNet	60	26	2,406	216.41	0.23
Mobile Net-V2	3.5	155	1,454	12.71	0.97
ResNet-50	25.5	178	1,746	91.04	1.41
Multi-input	15.5	86	181,214	91.52	1.36
Con1D	4.6	20	827	16.82	0.15
LSTM	3.2	12	955	12.49	0.026

one-dimensional hue illustrates the color characteristics of the FPBs for each grade. Fig. 5 shows the initial segment of the x-axis, which ranges from 10° to 90°, referring to the illustration of ripeness features correlated with the red or orange hues. The y-axis shows a noticeable peak value within the 50° to 70° range. Second, in the 120° to 180° range the degree of likelihood may be used to measure the unripe in some varieties of FPBs.

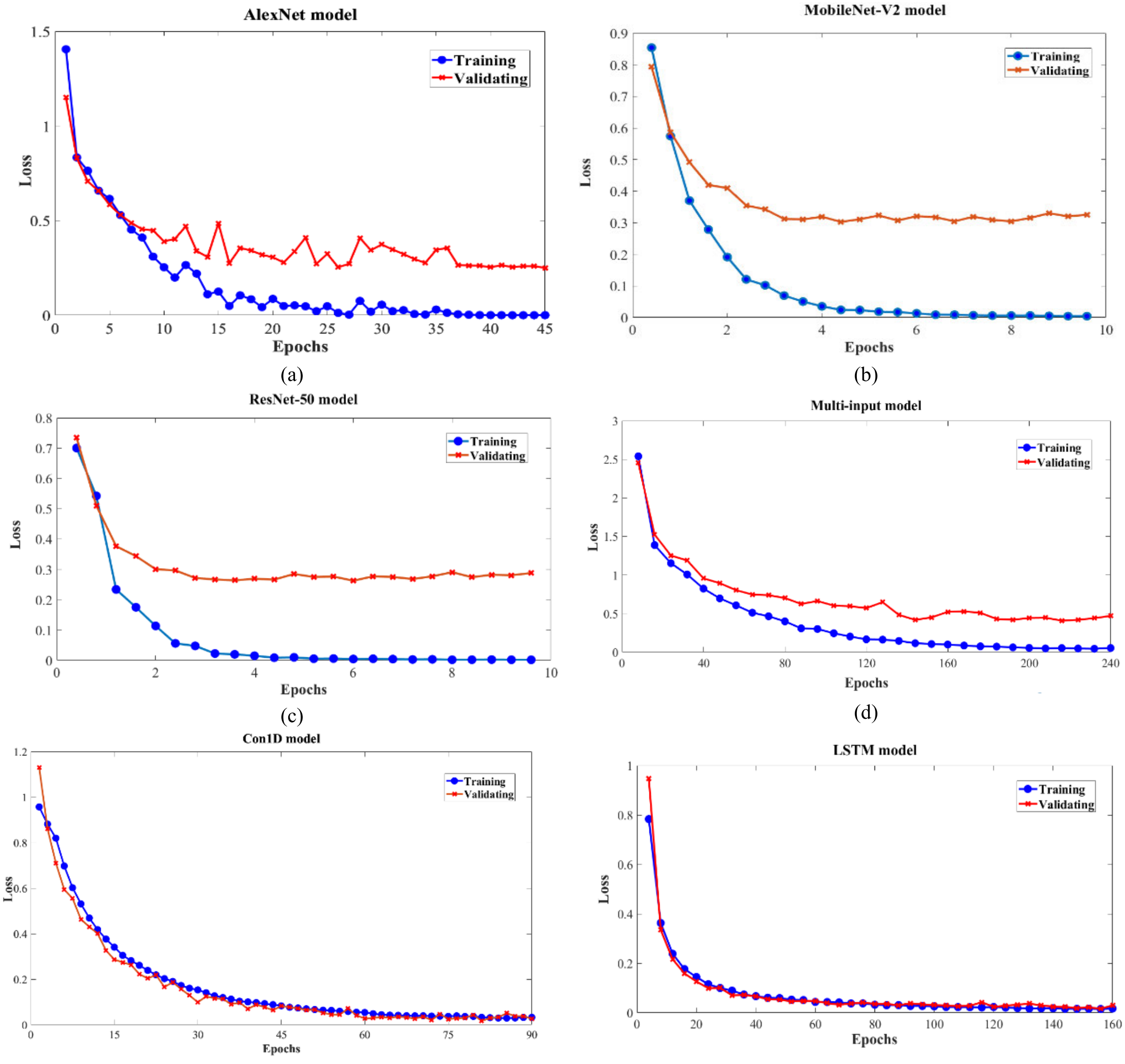
Noise-added data were prepared to evaluate the robustness of the model in actual environments. In addition, data augmentation was performed to avoid overfitting and underfitting. Fig. 6 illustrates the one-dimensional hue datasets for various SNR values ranging from 0 to 20 dB. The x-axis indicates the degree within the0 to 360°, whereas the y-axis represents the normalized intensity within the range of 0.0 to 1.0. The peak values occurred at 50° to 60° for a no-noise environment. At 0 dB, the peak at 60° vanished; at 5 dB, a spurious peak appeared at 10°. These two elements can be recognized in the graph, and noise is introduced, which modifies the visual representation of the hue intensity plot.

In the third stage, the DT-CWT technique is implemented using the "*dualtree*" MATLAB function. We propose implementing a one-level DT-CWT; hence, the output data size is half the size of the DT-CWT input. The dimension of the input dataset is 1 × 360, so the dimension of the decimator cascaded with a digital low-pass filter is 1 × 180. The summation of the real and imaginary channels was applied to highlight the important features with enhanced clarity, as shown in Fig. 7.

At 0 dB SNR, the peak at 30° occurred, as shown in Fig. 7. Before using the DT-CWT method, noise contaminates the features in a particular range of approximately 60°, as illustrated in Fig. 6. However, it is obvious that the frequency ranges from 100° to 180° presents an unusual amount of noise when increasing the SNR within the range of 0 to 5 dB. A possible impact of This may influence the complexity of the training process. Hence, it is necessary to perform data augmentation by adding a noise component that remains at a threshold of less than 10% of the original signal. This can be achieved by implementing an SNR range of 10 to 20 dB throughout the data augmentation.

The output of the DT-CWT is applied to the LSTM. Initially, the development of an LSTM model began with utilization the "*deep network designer*" and "*dlnet*" functions in MATLAB. This function describes the number of learnable layers in which the total number of learnable elements will impact both the processing time and the model's size. The LSTM model was compared with five other deep learning models: AlexNet, MobileNet-V2, ResNet-50, Multi-input [24], and Con1D [32]. The model's development process comprised three parts: training, validating, and testing. The ratios of the train-validation-test split of the datasets were 70%, 25%, and 5%, respectively. These ratios were obtained through a separate optimization. Utilization of the "*trainingPartitions*" function, which aims to improve the natural bias among the datasets, is provided. Additionally, the optimization technique was implemented in stochastic gradient descent with momentum (SGDM) configuration.

The performance results are listed in Table 1. The performance of the parameters and time training process describes the development of all networks compared to the learnable parameters, layer, training time, model sizes, and testing times. As shown in Table 1, the LSTM model outperformed



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 8. Loss convergence plot obtained from (a) AlexNet, (b) MobileNet-V2, (c) ResNet-50, (d) Multi-input, (e) Con1D, and (f) LSTM.

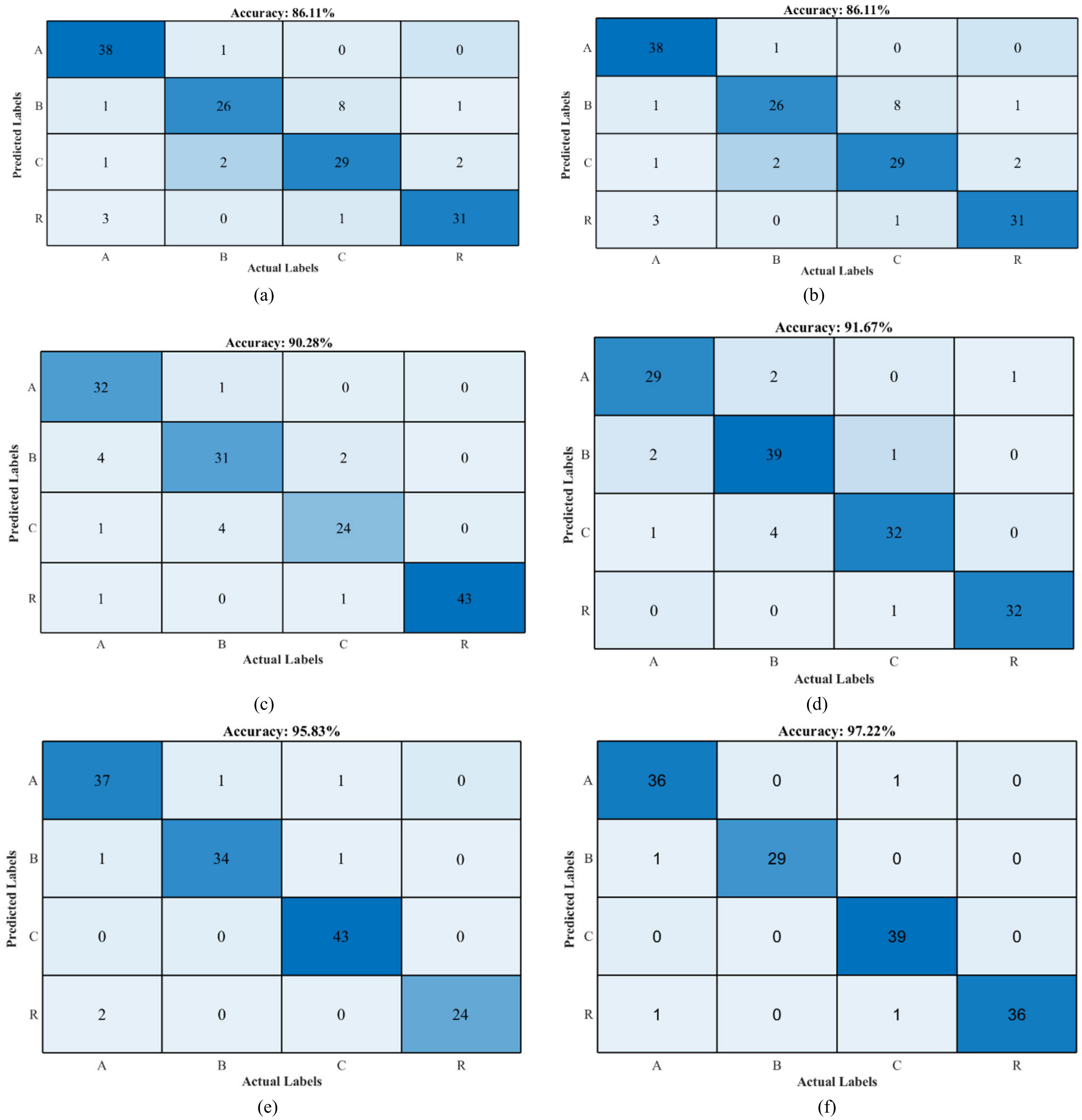
the other models for all parameters except the training time. The number of learnable parameters was 3.2 million, the number of layers was 12, the training time was 955 s, the model size was 12.49 MB, and the average testing time was 0.026 s. The Con1D model was the second-best model. It achieved the lowest training time, a comparably small model size (16.82 MB), and a very short testing time. The Con1D model consisted of 20 layers [32]. Another compact model is mobileNet-V2, which contains 3.5 million parameters and requires 12.71 MB for model storage. The number of layers is 155 higher than that of Con1D and LSTM. Hence, its training and testing times are longer than the other two best models.

Table 2 presents the training process, wherein the performance is compared based on various measurements, such as training loss, validation loss, validation accuracy, and the number of epochs achieved by different models.

TABLE 2. Training performances of models.

Model	Training loss	Validation loss	Validation accuracy (%)	Epochs
AlexNet	0.0001	0.245	93.62	40
MobileNet-V2	0.0011	0.3228	90.37	6
ResNet-50	0.0009	0.2813	92.44	5
Multi-input	0.0534	0.4699	90.26	200
Con1D	0.0303	0.0264	99.23	90
LSTM	0.0354	0.3891	99.51	100

The validation accuracy describes the accuracy of the model after completing the training process. Hence, the



Accuracy: 86.11%

	A	B	C	R
A	38	1	0	0
B	1	26	8	1
C	1	2	29	2
R	3	0	1	31

(a)

Accuracy: 86.11%

	A	B	C	R
A	38	1	0	0
B	1	26	8	1
C	1	2	29	2
R	3	0	1	31

(b)

Accuracy: 90.28%

	A	B	C	R
A	32	1	0	0
B	4	31	2	0
C	1	4	24	0
R	1	0	1	43

(c)

Accuracy: 91.67%

	A	B	C	R
A	29	2	0	1
B	2	39	1	0
C	1	4	32	0
R	0	0	1	32

(d)

Accuracy: 95.83%

	A	B	C	R
A	37	1	1	0
B	1	34	1	0
C	0	0	43	0
R	2	0	0	24

(e)

Accuracy: 97.22%

	A	B	C	R
A	36	0	1	0
B	1	29	0	0
C	0	0	39	0
R	1	0	1	36

(f)

FIGURE 9. Confusion matrix obtained from (a) AlexNet, (b) MobileNet-V2, (c) ResNet-50, (d) Multi-input, (e) Con1D, and (f) LSTM.

model presents an important degree of accuracy in its classification capability. The gap between the training and validation loss curves indicates overfitting or underfitting. Table 2 presents the training performance results. The LSTM model achieved a small gap of approximately 0.0028 between the training and validation of the loss. This resulted in a high validation accuracy of 99.66% with 100 epochs. Although the epoch numbers of MobileNet-V2 and RestNet-50 are very small compared to those of LSTM, their validation accuracy values are much lower, as shown in Fig. 8.

A comparative evaluation of the classification performances from the six deep learning models is presented, highlighting key metrics, as shown in Table 3. The LSTM model offered the best efficiency, that is, 99.62% accuracy,

TABLE 3. Comparison of classification performance of six deep learning models.

Model	Acc.	Pre.	Sen.	Spec.	F1 score	AUC
AlexNet	94.81	94.70	94.71	98.29	94.81	99.40
MobileNet-V2	90.72	90.81	90.94	96.90	90.72	97.88
ResNet-50	90.19	90.03	90.03	96.75	90.01	98.05
Multi-input	90.26	89.86	96.72	89.54	89.69	96.94
Con1D	99.42	99.43	99.38	99.81	99.42	99.99
LSTM	99.62	99.60	99.65	99.87	99.62	99.98

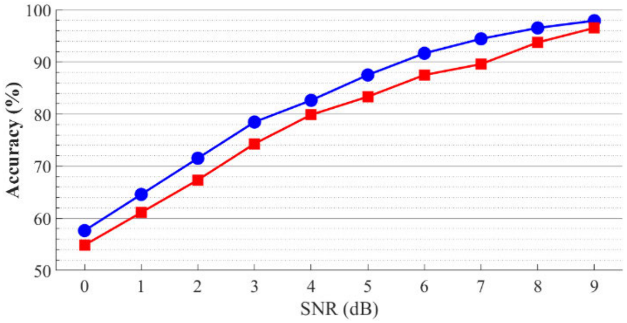


FIGURE 10. Performances of LSTM (blue) compared with Con1D (red) for different SNRs.

99.60% precision, 99.65% sensitivity, 99.87% specificity, 99.62% F1-score, and 99.98% AUC. The highest area under the highest AUC (99.99%) was obtained using the Con1D model.

The Con1D model was the second-best model for these key metrics. It achieved an accuracy of 99.42%, precision of 99.43%, sensitivity of 99.38%, specificity of 99.81%, and F1-score of 99.42%.

This section illustrates the primary purpose of the confusion matrix, which visually represents the relationship between actual and predicted labels. We used a confusion matrix as shown in Fig. 9 to demonstrate the model performance for classifying the four grades of FPBs. In our experiments, a total of 144 datasets were utilized in the testing process. These datasets were randomly determined and independent of the trained and validated datasets. The LSTM model achieved the highest accuracy of classification at 97.22%. In contrast, the Con1D model was second best, achieving a lower accuracy of 95.83%.

Upon evaluating the model performance in terms of accuracy and processing time efficiency, the LSTM and Con1D model demonstrated the highest level of accuracy and the shortest testing time. Next, the robustness of the model to noise was tested in several SNR environments, as shown in Fig. 10. The x-axis describes the various implemented SNR collection values between 0 dB and 9 dB. This is because the minimum noise power represents the actual scenario in which a noisy environment can be applied. The x-axis illustrates the accuracy of the testing in the evaluated datasets. The y-axis represents the accuracy of the test process. We set the accuracy threshold to approximately 90%. In particular, the robustness of the proposed work was shown in Fig. 10. The performance of the LSTM model was represented by the blue circle points, whereas that of the Conv1D model was illustrated by the red square points.

Figure 10 illustrates that the accuracy performance of both the LSTM and Con1D models dropped when SNR decreased. Very high accuracy of 94.44%, 96.53%, and 97.92% at SNR levels of 7, 8, and 9 dB are achieved from the LSTM model. The LSTM is superior in both accuracy and noise robustness. It is shown that the LSTM model achieves 91.67% at 6 dB SNR while the Con1D model requires more than 7 dB SNR for the same accuracy.

IV. DISCUSSION AND CONCLUSION

This study converted the image datasets to a one-dimensional hue histogram, and DT-CWT was utilized for denoising. An LSTM model was proposed. In particular, the proposed model accuracy was evaluated and compared to other models. The results indicated that the classification accuracy of the LSTM model achieved about 99.6% without any noise introduced to the experiment. Moreover, this model achieved an accuracy as high as 91.67% when the SNR was at 6 dB. With a limited number of parameters and layers in the computational transformation within the LSTM model, the utilization of processing resources is comparatively low. Therefore, the proposed model required less computation time while achieving a higher accuracy performance. In particular, this LSTM model used only 12 million parameters, resulting in a model size of 12.49 MB and a testing time of only 0.026 seconds while the other pre-trained models contain 60 million parameters, 216.41 MB, and 1.41 seconds per sample for testing. Though the proposed model is a promising solution, it is not ready to install and use since the model is programmed with MATLAB^{TM} which requires a legal license. Therefore, we plan to implement the system in a cheaper computing device and programming with Python language. It is important to remember that platform resource consumption is a significant challenge. The practicality of the platform is complex and limited. When selecting a trading system, it is essential to carefully evaluate the inclusion of applications that provide capabilities such as low maintenance requirements, cost-effectiveness, and minimal installation spaces.

REFERENCES

[1] Suharjito, G. N. Elwirehardja, and J. S. Prayoga, "Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches," *Comput. Electron. Agricult.*, vol. 188, Sep. 2021, Art. no. 106359.

[2] A. Septiarini, H. Hamdani, H. R. Hatta, and A. A. Kasim, "Image-based processing for ripeness classification of oil palm fruit," in *Proc. 5th Int. Conf. Sci. Inf. Technol. (ICSITech)*, Oct. 2019, pp. 23–26.

[3] A. Septiarini, A. Sunyoto, H. Hamdani, A. A. Kasim, F. Utaminingrum, and H. R. Hatta, "Machine vision for the maturity classification of oil palm fresh fruit bunches based on color and texture features," *Scientia Horticulturae*, vol. 286, Aug. 2021, Art. no. 110245.

[4] N. Fadilah, J. Mohamad-Saleh, Z. A. Halim, H. Ibrahim, and S. S. Ali, "Intelligent color vision system for ripeness classification of oil palm fresh fruit bunch," *Sensors*, vol. 12, no. 10, pp. 14179–14195, Oct. 2012.

[5] B. Van de Poel, I. Bulens, M. L. A. T. M. Hertog, L. Van Gastel, M. P. De Proft, B. M. Nicolai, and A. H. Geeraerd, "Model-based classification of tomato fruit development and ripening related to physiological maturity," *Postharvest Biol. Technol.*, vol. 67, pp. 59–67, May 2012.

[6] M. Makky and P. Soni, "Development of an automatic grading machine for oil palm fresh fruits bunches (FFBs) based on machine vision," *Comput. Electron. Agricult.*, vol. 93, pp. 129–139, Apr. 2013.

[7] C. L. Chew, B. A. Tan, J. Y. S. Low, N. I. N. M. Hakimi, S. F. Kua, and C. M. Lim, "Exogenous ethylene application on postharvest oil palm fruit bunches improves crude palm oil quality," *Food Sci. Nutrition*, vol. 9, no. 10, pp. 5335–5343, Aug. 2021.

[8] K. Tan, W. S. Lee, H. Gan, and S. Wang, "Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features in outdoor scenes," *Biosyst. Eng.*, vol. 176, pp. 59–72, Dec. 2018.

[9] B. Pamornnak, S. Limsiroratana, T. Khaorapapong, M. Chongcheawchamnan, and A. Ruckelshausen, "An automatic and rapid system for grading palm bunch using a Kinect camera," *Comput. Electron. Agricult.*, vol. 143, pp. 227–237, Dec. 2017.

[10] M. Rhif, A. Ben Abbes, I. Farah, B. Martínez, and Y. Sang, "Wavelet transform application for/in non-stationary time-series analysis: A review," *Appl. Sci.*, vol. 9, no. 7, p. 1345, Mar. 2019.

[11] L. Pan, P. Pipitsunthonsan, P. Zhang, C. Daengngam, A. Booranawong, and M. Chongcheawchamnan, "Noise reduction technique for Raman spectrum using deep learning network," in *Proc. 13th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2020, pp. 159–163.

[12] D. Min, Z. Jiuwen, and M. Yide, "Image denoising via bivariate shrinkage function based on a new structure of dual contourlet transform," *Signal Process.*, vol. 109, pp. 25–37, Apr. 2015.

[13] Z. Moravej, S. H. Mortazavi, and S. M. Shahrtash, "DT-CWT based event feature extraction for high impedance faults detection in distribution system," *Int. Trans. Electr. Energy Syst.*, vol. 25, no. 12, pp. 3288–3303, Dec. 2014.

[14] O. S. Faragallah, H. El-Hoseny, W. El-Shafai, W. A. El-Rahman, H. S. El-Sayed, E. M. El-Rabaie, F. E. A. El-Samie, and G. G. N. Geweid, "A comprehensive survey analysis for present solutions of medical image fusion and future directions," *IEEE Access*, vol. 9, pp. 11358–11371, 2021.

[15] W.-L. Zheng, W. Liu, Y. Lu, B.-L. Lu, and A. Cichocki, "EmotionMeter: A multimodal framework for recognizing human emotions," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1110–1122, Mar. 2019.

[16] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 123–151, Nov. 2005.

[17] R. Sammouda, A. M. S. Al-Salman, A. Gumaei, and N. Tagoug, "An efficient image denoising method for wireless multimedia sensor networks based on DT-CWT," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 11, Nov. 2015, Art. no. 632568.

[18] S. Chakraborty, A. Chatterjee, and S. K. Goswami, "A dual-tree complex wavelet transform-based approach for recognition of power system transients," *Expert Syst.*, vol. 32, no. 1, pp. 132–140, Jan. 2014.

[19] M.-P. Hosseini, A. Hosseini, and K. Ahi, "A review on machine learning for EEG signal processing in bioengineering," *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 204–218, 2021.

[20] Pooja, S. Pahuja, and K. Veer, "Recent approaches on classification and feature extraction of EEG signal: A review," *Robotica*, vol. 40, no. 1, pp. 77–101, May 2021.

[21] X. Yuan, C. Chen, X. Lei, Y. Yuan, and R. M. Adnan, "Monthly runoff forecasting based on LSTM-ALO model," *Stochastic Environ. Res. Risk Assessment*, vol. 32, no. 8, pp. 2199–2212, May 2018.

[22] S. Christin, É. Hervet, and N. Lecomte, "Applications for deep learning in ecology," *Methods Ecol. Evol.*, vol. 10, no. 10, pp. 1632–1644, Jul. 2019.

[23] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016, *arXiv:1610.02357*.

[24] P. Pipitsunthonsan, L. Pan, S. Peng, T. Khaorapapong, S. Nakasathien, S. Channumsin, and M. Chongcheawchamnan, "Palm bunch grading technique using a multi-input and multi-label convolutional neural network," *Comput. Electron. Agricult.*, vol. 210, Jul. 2023, Art. no. 107864.

[25] V. Rupapara, F. Rustam, H. F. Shahzad, A. Mehmood, I. Ashraf, and G. S. Choi, "Impact of SMOTE on imbalanced text features for toxic comments classification using RVVC model," *IEEE Access*, vol. 9, pp. 78621–78634, 2021.

[26] A. Demir, T. Koike-Akino, Y. Wang, M. Haruna, and D. Erdogmus, "EEG-GNN: Graph neural networks for classification of electroencephalogram (EEG) signals," in *Proc. 43rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Nov. 2021, pp. 1061–1067.

[27] N. Feng, S. Xu, Y. Liang, and K. Liu, "A probabilistic process neural network and its application in ECG classification," *IEEE Access*, vol. 7, pp. 50431–50439, 2019.

[28] Md. M. Islam and Md. M. H. Shuvo, "DenseNet based speech imagery EEG signal classification using gramian angular field," in *Proc. 5th Int. Conf. Adv. Electr. Eng. (ICAEE)*, Sep. 2019, pp. 149–154.

[29] A. J. Prakash and S. Ari, "AAMI standard cardiac arrhythmia detection with random forest using mixed features," in *Proc. IEEE 16th India Council Int. Conf. (INDICON)*, Dec. 2019, pp. 1–4.

[30] Y. Wang, Z. He, and Y. Zi, "Enhancement of signal denoising and multiple fault signatures detecting in rotating machinery using dual-tree complex wavelet transform," *Mech. Syst. Signal Process.*, vol. 24, no. 1, pp. 119–137, Jan. 2010.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[32] R. Yang, S. K. Singh, M. Tavakkoli, N. Amiri, Y. Yang, M. A. Karami, and R. Rai, "CNN-LSTM deep learning architecture for computer vision-based modal frequency detection," *Mech. Syst. Signal Process.*, vol. 144, Oct. 2020, Art. no. 106885.

[33] D. Yu, H. Wang, P. Chen, and Z. Wei, "Mixed pooling for convolutional neural networks," in *Rough Sets and Knowledge Technology*. Springer, 2014, pp. 364–375.

[34] C.-Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree," 2015, *arXiv:1509.08985*.

[35] S. R. Nayak, D. R. Nayak, U. Sinha, V. Arora, and R. B. Pachori, "Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study," *Biomed. Signal Process. Control*, vol. 64, Feb. 2021, Art. no. 102365.

[36] Z. Yue, L. Ma, and R. Zhang, "Comparison and validation of deep learning models for the diagnosis of pneumonia," *Comput. Intell. Neurosci.*, vol. 2020, pp. 1–8, Sep. 2020.

[37] R. C. Prati, G. E. A. P. A. Batista, and M. C. Monard, "A survey on graphical methods for classification predictive performance evaluation," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 11, pp. 1601–1618, Nov. 2011.

PRONTHEP PIPITSUNTHONSAN was born in Songkhla, Thailand, in 1989. He received the bachelor's and master's degrees in computer engineering from the Prince of Songkla University, in 2010 and 2017, respectively. He is currently pursuing the Ph.D. degree in electrical engineering. He has work experience with the Geo-Informatics and Space Technology Development Agency (GISTDA), from 2015 to 2017. His research interests include deep learning, big data, and agriculture technology.

THANATE KHAORAPAPONG was born in Thailand, in 1967. He received the bachelor's degree in electrical engineering from the Prince of Songkla University, in 1990, and the D.E.A. degree in automatism and the Ph.D. degree in doctorate system automatiques from INPT/ENSEEIHT, France, in 1998 and 2001, respectively. He is currently a Professor of computer engineering with the Prince of Songkla University. His research interests include image processing, robotics, and control systems.

SUTKHET NAKASATHIEN received the Ph.D. degree in crop science from North Carolina State University, USA, in 1998. He is highly respected for his expertise in the fields of agriculture and food and is widely recognized for his extensive senior management experience. He was the former Dean of the Faculty of Agriculture, Kasetsart University, from 2013 to 2020. At present, he has served as the President of the Research University Network (RUN), Thailand, and the President of the Agriculture Science Society of Thailand under the Patronage of His Majesty the King. Currently, he works as a Professor at the Faculty of Agriculture, Kasetsart University, Thailand.

PETCHARAT SURIYACHAI received the B.S. and M.S. degrees in electrical and computer engineering from Carnegie Mellon University, USA, and the Ph.D. degree in computer science from Lancaster University, U.K. After graduation, she became an Assistant Professor with the Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Thailand. Her research interests include the Internet of Things, artificial intelligence, wireless sensor networks, communication technologies, and computer networks.

SITTIPORN CHANNUMSIN received the bachelor's degree in electronics engineering from the King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2006, the M.Sc. degree in space technology and planetary exploration from the University of Surrey, in 2011, and the Ph.D. degree in aerospace engineering from the University of Glasgow, U.K., in 2016. He is currently a Supervisor of the Astrodynamics Research Laboratory (AstroLab), Geo-Informatics and Space Technology Development Agency (GISTDA). His research interests include spaceflight dynamics, space debris mitigation, machine learning, and orbit and attitude control and optimization. He serves as the co-editor and a technical reviewer.

MITCHAI CHONGCHEAWCHAMNAN (Senior Member, IEEE) was born in Bangkok, Thailand. He received the B.Eng. degree in telecommunication from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, in 1992, the M.Sc. degree in communication and signal processing from Imperial College London, London, U.K., in 1995, and the Ph.D. degree in electrical engineering from the University of Surrey, Guildford, U.K., in 2001. He joined the Mahanakorn University of Technology, Bangkok, as a Lecturer, in 1992. He is currently a Professor with the Faculty of Engineering, Prince of Songkla University, Songkhla, Thailand. His current research interests include deep learning, microwave circuit design, and 3D printing techniques for millimeter-wave circuits.