A Hybrid Deep Learning and Optimized Machine Learning Approach for Rose Leaf Disease Classification

Sumitra Nuanmeesri
Department of Information Technology
Faculty of Science and Technology
Suan Sunandha Rajabhat University
Bangkok, Thailand
sumitra.nu@ssru.ac.th

Abstract-Analysis of the symptoms of rose leaves can identify up to 15 different diseases. This research aims to develop Convolutional Neural Network models for classifying the diseases on rose leaves using hybrid deep learning techniques with Support Vector Machine (SVM). The developed models were based on the VGG16 architecture and early or late fusion techniques were applied to concatenate the output from a fully connected layer. The results showed that the developed models based on early fusion performed better than the developed models on either late fusion or VGG16 alone. In addition, it was found that the models using the SVM classifier had better efficiency in classifying the diseases appearing on rose leaves than the models using the softmax function classifier. In particular, a hybrid deep learning model based on early fusion and SVM, which applied the categorical hinge loss function, yielded a validation accuracy of 88.33% and a validation loss of 0.0679, which were higher than the ones of the other models. Moreover, this model was evaluated by 10-fold cross-validation with 90.26% accuracy, 90.59% precision, 92.44% recall, and 91.50% F1-score for disease classification on rose leaves.

Keywords-hybrid deep learning; neural networks; rose leaf diseases; support vector machine

I. INTRODUCTION

Roses are widely produced and exported globally. In 2019, the export value of roses was more than 175 million US dollars. The top five countries with the highest export rankings are Netherlands, Denmark, Uganda, Germany, and Canada [1]. In cultivating roses, pest problems such as insect infestations are often encountered along with pathogens caused by fungi, viruses, and bacteria [2]. There are also diseases caused by nutritional deficiencies such as in nitrogen, iron, zinc, and magnesium. Disease symptoms can be detected in roots, stems, branches, leaves, and buds or flowers. Especially the leaves are a source of various infectious disease symptoms. However, classifying an infected disease requires skill and experience. For example, rose mosaic disease is a common disease worldwide and can sometimes be caused by more than one pathogen [3]. Image processing methods for plant disease classification are currently being studied [4] combined with machine learning such as Support Vector Machine (SVM) [5] or K-Nearest Neighbors (KNN) [6]. For example, the authors in [7] classified 4 rose leaf diseases using machine learning with with at least 94% accuracy. In addition to machine learning, other methods such as deep learning and neural networks are applied to recognize and classify plant diseases. The author in [8] developed a Convolutional Neural Network (CNN) model, which applied MobileNet and transfer learning for rose disease classification. Over 30 and at least 15 rose diseases can be observed on the leaves [2].

Most of the research used a single perspective or a single set of images as the dataset for model training. However, in the deep learning model training, it is necessary to use images with multiple perspectives, such as image augmentation, and include image segmentation to highlight features that appear on the image. This approach usually increases the accuracy of the model. Moreover, there are currently very few studies that have applied hybrid deep learning models to classify plant diseases. Therefore, this research aims to develop rose leaf disease classification models using hybrid deep learning. Besides, this work also compares the performance of models using different classifiers, namely softmax function and SVM.

II. RESEARCH METHODOLOGY

A. Image Data Collection

This study classifies rose diseases by identifying the symptoms on leaves based on image processing and CNN. A program was developed using the Google search engine and ChromeDriver utility to search and download rose leaf images with dimensions of at least 224 pixels. All downloaded images were rechecked and labeled. Moreover, the author took photos of rose leaves with and without diseases with an Android mobile phone. Therefore, a dataset of 4,032 downloaded and taken pictures was formed. The imagres were categorized to 16 different classes with regard to the shown rose disease (15 diseases + 1 normal/control) as shown in Table I. Finally, all images were resized and cropped to the dimension of 224×224 pixels.

Corresponding author: Sumitra Nuanmeesri

TABLE I. ROSE LEAF IMAGES CLASSIFICATION

No.	Label	Disease	Caused by	Amount (images)
1	BS	Black spot	Fungi	295
2	DM	Downy mildew	Fungi	272
3	PM	Powdery mildew	Fungi	281
4	BM	Black mold	Fungi	245
5	BB	Botrytis blight	Fungi	234
6	VW	Verticillium wilt	Fungi	220
7	CLS	Cercospora leaf spot	Fungi	279
8	ATN	Anthracnose	Fungi	276
9	RT	Rust	Fungi	280
10	RM	Rose mosaic	Virus	270
11	RR	Rose Rosette	Virus	236
12	RSD	Rose spring dwarf	Virus	281
13	SLR	Strawberry latent ringspot	Virus	183
14	IB	Insect bites	Insect	192
15	ND	Nutrient deficiencies	-	238
16	NM	Normal (disease-free)	-	250

B. Image Augmentation

During the deep learning training mstage many images are needed to increase the performance of the model. If a particular class has a small number of images, it can affect classification accuracy. Thus, the author increased the number of training images by using the image augmentation technique, including vertical and horizontal flips, rotation (45, -45, 90, -90 degrees), shearing (45 and -45 degrees), and random zoom-in up to 200%. Thus, the number of the images increased from 4,032 to 40,320 images. The example of image augmentation is shown in Figure 1.

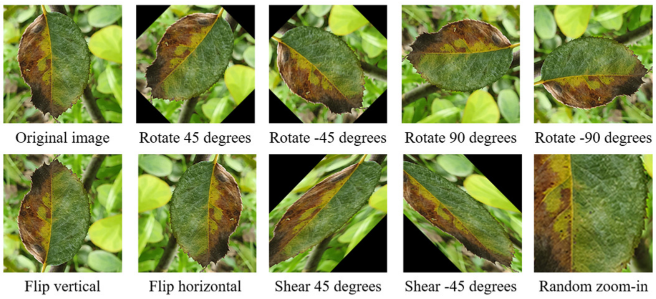
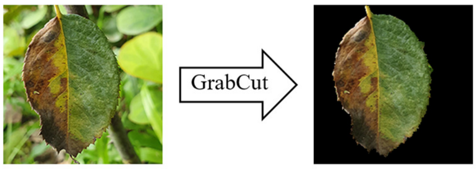

Original image
Rotate 45 degrees
Rotate -45 degrees
Rotate 90 degrees
Rotate -90 degrees
Flip vertical
Flip horizontal
Shear 45 degrees
Shear -45 degrees
Random zoom-in

Fig. 1. Example of image augmentation.

C. Image Processing

Image processing was applied to emphasize the physical appearance of the rose leaves. The preliminary step is that the rose leaf was separated from the background pixels with the GrabCut [9] method based on the graph-cut technique. Such an output image is illustrated in Figure 2.



Original image
Image without background pixels

Fig. 2. The background pixels are removed by the GrabCut method.

The images obtained after the removal of the background pixels were subjected to image color-spacing and image thresholding processing.

1) Hue, Saturation, Value Color Space

Hue, Saturation, Value (HSV) is a color space model which includes ranges of color type between 0 to 360 degrees, vibrancy, and color brightness. This work focused on color ranges around 120 degrees, which are related to the green color of a rose leaf. All green areas were desaturated with the lower saturation as grayness. As a result, any color not related to green was accelerated to become more emphatic.

2) Truncated Adaptive Gaussian Thresholding

Truncated Adaptive Gaussian Thresholding (TAGT) is a combination technique between truncate and adaptive Gaussian thresholding. For truncate thresholding, an image without background pixels is processed. Pixels greater than the threshold value th_{val} = 127, were assigned that value [10]:

$$out(x,y) = \begin{cases} \text{threshold}, & \text{if } in(x,y) > th_{val} \\ in(x,y), & \text{otherwise} \end{cases} \quad (1)$$

where $in(x,y)$ refers to the input pixel coordination, and $out(x,y)$ refers to the output pixel coordination.

Next, the threshold was adjusted from the weighted sum of the block size of the pixel neighborhood at 7×7 using adaptive Gaussian and binary thresholding, as in (2) [10].

$$out(x,y) = \begin{cases} max_{val}, & \text{if } in(x,y) > th(x,y) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where max_{val} refers to the maximum value assigned to the pixels and $th(x,y)$ refers to the individual threshold calculation of each pixel.

3) Double Inverse-Binary Thresholding

Double Inverse-Binary Thresholding (DIBT) is a thresholding method with twice applied inverse-binary thresholding. First, the image without background pixels is taken through a thresholding process between inverse-binary and binary, where the threshold values are set to 100 and 0 respectively. The resulting image from the first step was adjusted to the threshold value of 127 by inverse-binary thresholding, calculated in (3) [10].

$$out(x,y) = \begin{cases} 0, & \text{if } in(x,y) > th(x,y) \\ max_{val}, & \text{otherwise} \end{cases} \quad (3)$$

This results in an image emphasizing the expected coordinates of the suspected position of disease infection or wilt on the rose leaf. Each original image will result to 3 more images, namely HSV, TAGT, and DIBT, as shown in Figure 3.

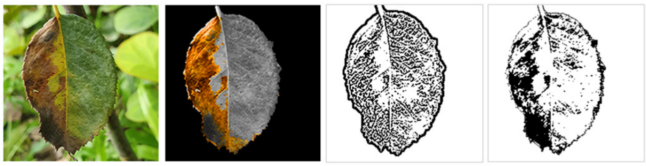

Original
HSV
TAGT
DIBT

Fig. 3. Example of each image processing.

D. Hybrid Deep Learning Modeling

In this step the CNN models for the classification of diseases on a rose leaf were developed. Twelve models were developed as follows.

1) Visual Geometry Group16-Based CNN Model

Visual Geometry Group16 (VGG16) model is a CNN architecture presented in [11]. The input images of the VGG16-based CNN model were set to 224×224 pixels for processing through 16 weight layers, including 13 convolution layers and 3 fully connected layers. All convolution layers have a 3×3 kernel size, 1 pixel of padding size, and the Rectified Linear Unit (ReLU) activation function. Spatial pooling followed with 5 max-pooling layers with a 2×2 pixel filter and stride 2. Further, 1 flatten layer was included before feeding the output to the fully connected layers. Furthermore, the softmax activation function was applied with 1,000 output classes in the last fully connected layer. Thus, the total trainable parameters of this model were 138,357,544 as shown in Figure 4.

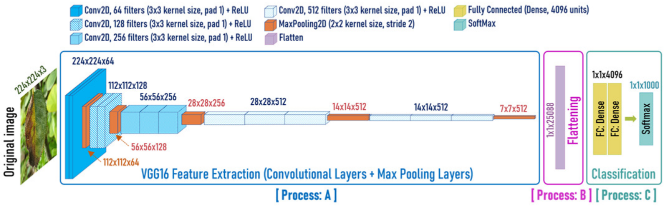

Conv2D, 64 filters (3x3 kernel size, pad 1) + ReLU
Conv2D, 128 filters (3x3 kernel size, pad 1) + ReLU
Conv2D, 256 filters (3x3 kernel size, pad 1) + ReLU
Conv2D, 512 filters (3x3 kernel size, pad 1) + ReLU
MaxPooling2D (2x2 kernel size, stride 2)
Flatten
Fully Connected (Dense, 4096 units)
SoftMax
Original image
224x224x3
224x224x64
112x112x128
56x56x256
28x28x256
28x28x512
14x14x512
14x14x512
7x7x512
56x56x128
112x112x64
VGG16 Feature Extraction (Convolutional Layers + Max Pooling Layers)
1x1x25088
Flattening
1x1x4096
1x1x1000
FC: Dense
FC: Dense
Softmax
Classification
[Process: A]
[Process: B]
[Process: C]

Fig. 4. VGG16-based CNN model architecture.

According to Figure 4, the "Process: A" refers to feature extraction layers, "Process: B" refers to the flatten layer, and "Process: C" refers to the fully connected layers. This research used the original image dataset with 16 labeled output classes (see Table I) to develop the VGG16-based CNN model. Therefore, the softmax layer (in Process: C) was set to 16 instead of 1000 classes. The overall trainable parameters of this model were 134,326,096.

2) Early Fusion Model

The early fusion (EF) model was developed based on the VGG16-based CNN model. It starts with the images that have undergone each image thresholding processing separately (original, HSV, TAGT, and DIBT images) to each channel of CNN for extracting features (see Process: A in Figure 4) with the VGG16 architecture. The outputs obtained for each dataset were fused and flattened before being classified with the fully connected layers, as shown in Figure 5. The sum of the trainable parameters was 484,041,973.

3) Late Fusion Model

The VGG16-based CNN model was extended to the Late Fusion (LF) model in this work. The LF model starts with each processed image as input (as the EF model) for each CNN channel, then fused each output obtained after classification by the softmax activation function. After the fusion of the results obtained from the 4 image types, they were classified by 2 dense layers of size 4,096 and were finalized with the softmax function to 16 output classes (Figure 6). The total trainable parameters were 586,665,136.

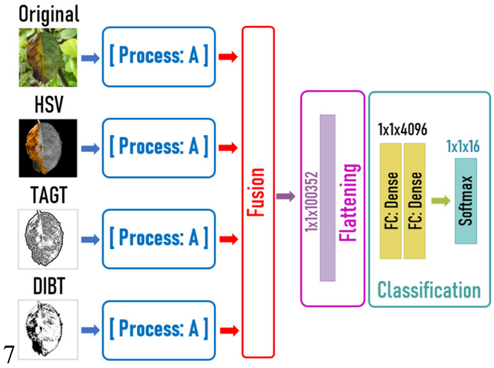


Fig. 5. VGG16-based early fusion model.

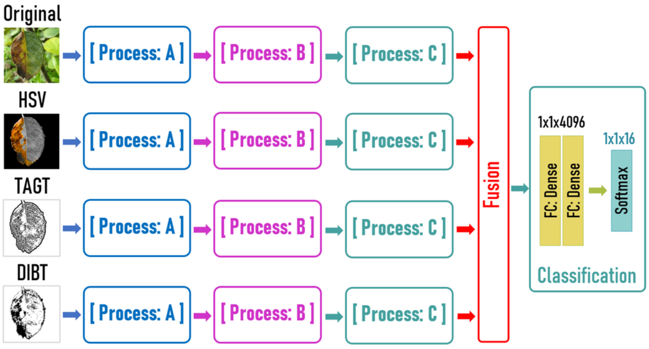


Fig. 6. VGG16-based late fusion model.

4) VGG16-Based SVM Models

According to the VGG16-based CNN model, the softmax activation function was used to classify the final output at the last layer. In contrast, this VGG16-based SVM model applied the SVM classifier instead of the softmax activation function, as shown in Figure 7.

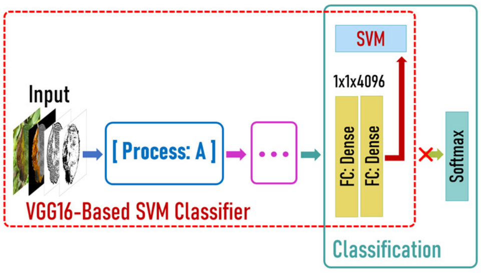


Fig. 7. VGG16-based SVM model.

SVM is a popular classifier for supervised learning algorithms, especially for binary classification. In this work, multi-class SVM is required to classify 16 classes of rose leaf disease images. There are several classifiers for multi-class SVM. In this work various multi-class SVM classifiers were applied including L2-SVM, Categorical Hinge Loss SVM (CHL-SVM), and Weston-Watkins SVM (WW-SVM), to the VGG16-based CNN models as follows.

The L2-SVM is based on the optimization of L2 norm and Squared Hinge Loss (SHL) which is calculated in (4) and (5).

$$y = w^T x_i + b \quad (4)$$

$$L = \sum_{i=1}^{N} \max(0, 1 - \hat{y}_i y)^2 \quad (5)$$

where w^T refers to the weight of dataset T, x_i refers to the augmentation of sample data vectors, b refers to the bias, N refers to the number of samples in a dataset, $\hat{y}$ refers to the actual class, and y refers to the predicted class.

Then, the SHL was optimized with a minimum of Euclidean norm and a large error penalty. Thus, the L2-SVM was formulated in (6).

$$\min \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^{N} \max(0, 1 - \hat{y}_i (w^T x_i + b))^2 \quad (6)$$

where $\|w\|_2$ refers to the Euclidean norm (L2 norm regularization), and C refers to the large error penalty for misclassification in which $C > 0$.

The CHL-SVM or multi-class hinge loss function was implemented with TensorFlow 2 based Keras and is calculated in (7) [12].

$$L = \max\left(0, 1 + \max\left((1 - \hat{y})y\right) - \sum_{i=1}^{N} \hat{y}_i y_i\right) \quad (7)$$

Regarding the WW-SVM or Weston-Watkins hinge loss [13], the linear classifier was calculated in (8), and the optimization was formulated in (9).

$$L = \sum_{i=1}^{l} \sum_{m \neq y_i} \max(0, 1 + y_i - y_m) \quad (8)$$

$$\min \frac{1}{2} \sum_{m=1}^{k} \|w_m\|^2 + C \sum_{i=1}^{l} \sum_{m \neq y_i} \max(0, 1 + y_i - y_m) \quad (9)$$

where w refers to the weight, k refers to the number of classes, l refers to the number of samples in the dataset, y refers to the predicted class, and C refers to the large error penalty for misclassification in which $C > 0$.

Finally, the last fully connected layer with the softmax function was replaced with the multi-class SVM to classify the final output. Therefore, 3 VGG16-based SVM models, namely VGG16 & L2-SVM, VGG16 & CHL-SVM, and VGG16 & WW-SVM, were developed with different multi-class SVM classifiers.

5) Early Fusion-Based SVM Models

The early fusion-based SVM model applied the 3 different multi-class SVM classifiers to the last fully connected layer with softmax of the EF model. Thus, 3 EF models were developed, namely EF & L2-SVM, EF & CHL-SVM, and EF & WW-SVM.

6) Late Fusion-Based SVM Models

The LF model included two softmax functions at two layers: before fusion and at the last layer of the model. Thus, the softmax classifier of the last layer was bypassed and replaced with the multi-class SVM classifiers. Therefore, 3 late fusion-based SVM models were developed: LF & L2-SVM, LF & CHL-SVM, and LF & WW-SVM.

By default, all models are based on VGG16 and CNN architecture. The image dataset was randomly split into 70%, 15%, and 15% for training, validating, and testing respectively. The hyperparameters were set as follows: the batch size was 64, the learning rate was 0.001, and training took 200 epochs. The models were compiled with the Adam optimizer.

E. Model Evaluation

All models were evaluated and validated by the accuracy and loss value during the training processing. In addition, the VGG16-based CNN, EF, and LF models that applied the softmax function at the fully connected layer were evaluated using the categorical cross-entropy loss [14]:

$$\text{Cross-entropy} = -\sum_{i=1}^{m} \sum_{j=1}^{n} y_{i,j} \log\left(p_{i,j}\right) \quad (10)$$

where m refers to the total of input, n refers to the number of classes, $y_{i,j}$ refers to the input i of class j, $p_{i,j}$ refers to the probability of the predicted class j by input i.

Further, the k-fold cross-validation was used to estimate the learning skill of the model based on an unseen dataset. The k-fold cross-validation is mainly used to measure performance for machine learning models but can also be applied to deep learning models. Thus, all 12 models were evaluated by 10-fold cross-validation in this work. The performance of the models was validated on accuracy (ACC) [15], precision (PREC) [16], recall (REC) [17], and F1-Score.

III. RESULTS

All 12 models were trained, validated, and tested and their performances were compared.

A. Model Training and Validation Performance

The results showed that the models developed with the early fusion technique performed better than late fusion and VGG16 models. Especially, the model developed with the early fusion method and categorical hinge loss for the SVM (EF & CHL-SVM) gave the best accuracy among the models as shown in Table II.

TABLE II. ACCURACY AND LOSS VALUES RESULTS BETWEEN TRAINING AND VALIDATION OF THE MODELS

Models	Training		Validation	
	ACC	Loss	ACC	Loss
VGG16 (Softmax)	94.71	0.1847	83.28	0.4063
VGG16 & L2-SVM	95.09	0.1656	83.79	0.3643
VGG16 & CHL-SVM	96.03	0.1185	84.32	0.2607
VGG16 & WW-SVM	95.59	0.1404	84.25	0.3090
EF (Softmax)	97.09	0.0654	87.31	0.1438
EF & L2-SVM	97.37	0.0514	87.62	0.1130
EF & CHL-SVM	98.28	0.0309	88.33	0.0679
EF & WW-SVM	97.91	0.0443	88.07	0.0974
LF (Softmax)	96.44	0.0981	84.19	0.2159
LF & L2-SVM	96.86	0.0770	84.87	0.1695
LF & CHL-SVM	97.21	0.0597	85.44	0.1313
LF & WW-SVM	96.97	0.0714	85.02	0.1570

According to Table II, the EF & CHL-SVM model yielded a training accuracy of 98.28% and a training loss of 0.0309. In addition, it gave a validation accuracy of 88.33% and a validation loss of 0.0679. The accuracy and loss error between training and validation of the EF & CHL-SVM are shown in Figure 8. The validation accuracy of the EF & CHL-SVM model for rose leaf disease classification can be normalized and displayed as a confusion matrix, as shown in Figure 9. The EF

& CHI-SVM model was able to classify the disease-free (NM) rose leaves with 98.95% accuracy. The most accurate classifications of rose leaf diseases were VW, IB, and DM, with 98.48%, 94.08%, and 92.68% accuracy respectively. For most of the other diseases the accuracy was higher than 87%, except for BB, ATN, CLS, and SLR which had less than 83%. Especially, the SLR disease had the lowest accuracy of 74.44%.

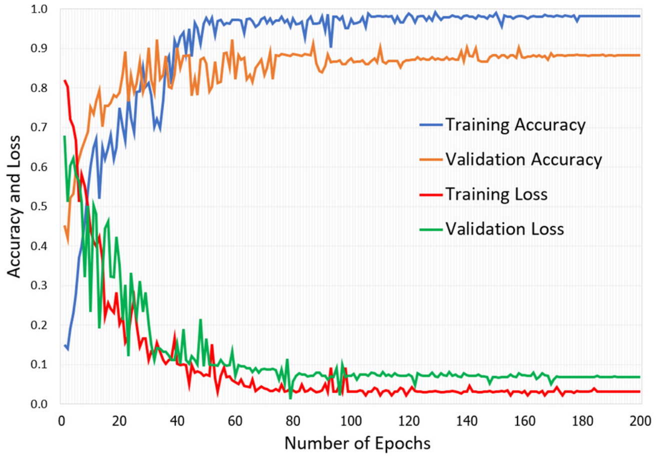
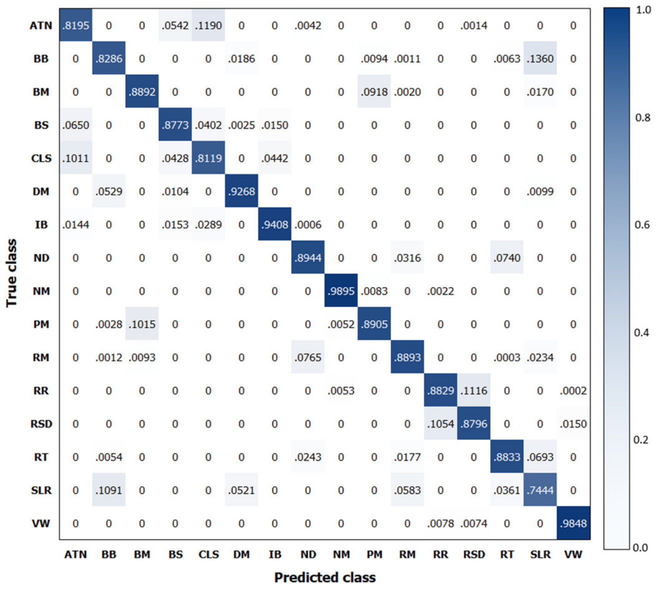


Fig. 8. The performance of training and validation of the EF & CHL-SVM model.



	ATN	BB	BM	BS	CLS	DM	IB	ND	NM	PM	RM	RR	RSD	RT	SLR	VW
ATN	.8195	0	0	.0542	.1190	0	0	.0042	0	0	0	0	.0014	0	0	0
BB	0	.8286	0	0	0	.0186	0	0	0	.0094	.0011	0	0	.0063	.1360	0
BM	0	0	.8892	0	0	0	0	0	0	.0918	.0020	0	0	0	.0170	0
BS	.0650	0	0	.8773	.0402	.0025	.0150	0	0	0	0	0	0	0	0	0
CLS	.1011	0	0	.0428	.8119	0	.0442	0	0	0	0	0	0	0	0	0
DM	0	.0529	0	.0104	0	.9268	0	0	0	0	0	0	0	0	.0099	0
IB	.0144	0	0	.0153	.0289	0	.9408	.0006	0	0	0	0	0	0	0	0
ND	0	0	0	0	0	0	0	.8944	0	0	.0316	0	0	.0740	0	0
NM	0	0	0	0	0	0	0	0	.9895	.0083	0	.0022	0	0	0	0
PM	0	.0028	.1015	0	0	0	0	0	.0052	.8905	0	0	0	0	0	0
RM	0	.0012	.0093	0	0	0	0	.0765	0	0	.8893	0	0	.0003	.0234	0
RR	0	0	0	0	0	0	0	0	.0053	0	0	.8829	.1116	0	0	.0002
RSD	0	0	0	0	0	0	0	0	0	0	0	.1054	.8796	0	0	.0150
RT	0	.0054	0	0	0	0	0	.0243	0	0	.0177	0	0	.8833	.0693	0
SLR	0	.1091	0	0	0	.0521	0	0	0	0	.0583	0	0	.0361	.7444	0
VW	0	0	0	0	0	0	0	0	0	0	0	.0078	.0074	0	0	.9848

Fig. 9. The validation accuracy of the EF & CHL-SVM model in a confusion matrix.

B. *Model Evaluation*

All developed models were tested and evaluated by 10-fold cross-validation with a test dataset. The result showed that the performances of the EF-based models were higher than the LF-based and VGG16-based models'. Regarding the EF-based models, the EF & CHL-SVM had the highest performance with 90.26% accuracy, 90.59% precision, 92.44% recall, and 91.50% F1-score as shown in Table III.

TABLE III. CROSS-VALIDATION PERFORMANCE

Models	ACC	PREC	REC	F1-Score
VGG16 (Softmax)	86.16	87.03	89.05	88.03
VGG16 & L2-SVM	86.37	87.27	89.16	88.20
VGG16 & CHL-SVM	86.90	87.81	89.47	88.63
VGG16 & WW-SVM	86.62	87.53	89.30	88.41
EF (Softmax)	89.54	90.20	91.90	91.04
EF & L2-SVM	89.85	90.51	92.14	91.32
EF & CHL-SVM	90.26	90.59	92.44	91.50
EF & WW-SVM	90.08	90.63	92.42	91.51
LF (Softmax)	87.62	88.56	90.04	89.30
LF & L2-SVM	87.83	88.78	90.19	89.48
LF & CHL-SVM	88.31	89.13	90.68	89.90
LF & WW-SVM	88.05	88.88	90.43	89.65

IV. CONCLUSION

This research developed 12 models for classifying rose diseases from the symptoms that appear on the rose leaves using a CNN model based on VGG16 architecture and image processing. The classification of rose diseases consists of 16 classes (9 classes for diseases caused by fungi, 4 for virus diseases, 1 for insect bit, 1 for nutrient deficiencies, and 1 disease-free class). The 12 developed CNN models were divided into three groups: VGG16, EF, and LF. In addition, each group was divided into two classifier types: softmax and SVM. The softmax function was used in 3 models, namely the VGG16-based CNN, EF, and LF models. The utilized multi-class SVM classifiers were L2-SVM, CHL-SVM, and WW-SVM. There were 4,032 rose leaf images for model training. The images were resized to 224×224 pixels and underwent image augmentation, resulting in a dataset of 40,320 images. These images were subjected to image processing, including removal of background pixels, HSV color space, TAGT, and DIBT, to emphasize their features. Both TAGT and DIBT are based on image thresholding processing. Ultimately, the dataset was split to 70% for training, 15% for validation, and 15% for model testing by 10-fold cross-validation.

The results showed that the EF-based models gave the highest training, validation, and testing performance values, followed by the LF-based and the VGG16-based models. In addition, the models developed with the SVM classifier performed higher than the models using the softmax function. The model using CHL-SVM showed the highest performance, followed by the models using WW-SVM, L2-SVM, and softmax function. Thus, the EF & CHL-SVM, a developed model based on the early fusion method and employing the SVM categorical hinge loss function was the most suitable model for classifying diseases on rose leaves with an accuracy of at least 88.33%. The models developed in [7, 8] had accuracy not less than 94%, which is higher than the accuracy of the CHL-SVM model in this work. However, these two studies only classified 4 rose leaf diseases, unlike this study which classified 15.

Moreover, it is evident that image processing can improve rose leaf disease classification accuracy, especially when the features are fused. Besides, it was found that SVM gave better results as a classifier than the softmax activation function, which is consistent with the findings in [18].

Regarding further work, the author plans to develop a model based on U-net deep learning and a transfer learning approach to detect and classify diseases on plants and then integrate it to the Internet of Things.

ACKNOWLEDGMENT

The author wishes to thank the Institute for Research and Development, Suan Sunandha Rajabhat University, for supporting this research.

REFERENCES

[1] "Top countries for Export of Roses," *NationMaster*. https://www.nationmaster.com/nmx/ranking/export-of-roses (accessed Sep. 21, 2021).

[2] R. K. Horst and R. A. Cloyd, *Compendium of Rose Diseases and Pests. Second Edition*, 2nd ed. St. Paul, MI, USA: The American Phytopathological Society, 2007.

[3] I. Vazquez-Iglesias *et al.*, "High throughput sequencing and RT-qPCR assay reveal the presence of rose cryptic virus-1 in the United Kingdom," *Journal of Plant Pathology*, vol. 101, no. 4, pp. 1171–1175, Nov. 2019, https://doi.org/10.1007/s42161-019-00307-5.

[4] S. Minaee, M. Jafari, and N. Safaie, "Design and development of a rose plant disease-detection and site-specific spraying system based on a combination of infrared and visible images," *Journal of Agricultural Science and Technology*, vol. 20, no. 1, pp. 23–36, Jan. 2018.

[5] D. Das, M. Singh, S. S. Mohanty, and S. Chakravarty, "Leaf Disease Detection using Support Vector Machine," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, Jul. 2020, pp. 1036–1040, https://doi.org/10.1109/ICCSP48568.2020.9182128.

[6] A. A. Bharate and M. S. Shirdhonkar, "A review on plant disease detection using image processing," in *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, Palladam, India, Dec. 2017, pp. 103–109, https://doi.org/10.1109/ISS1.2017.8389326.

[7] K. Swetharani and V. Prasad, "Design and Implementation of an Efficient Rose Leaf Disease Detection using K-Nearest Neighbours," *International Journal of Recent Technology and Engineering*, vol. 9, no. 3, pp. 21–27, Sep. 2020, https://doi.org/10.35940/ijrte.C4213.099320.

[8] A. Rajbongshi, T. Sarker, Md. M. Ahamad, and Md. M. Rahman, "Rose Diseases Recognition using MobileNet," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Istanbul, Turkey, Oct. 2020, https://doi.org/10.1109/ISMSIT50672.2020.9254420.

[9] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, Aug. 2004, https://doi.org/10.1145/1015706.1015720.

[10] "OpenCV: Miscellaneous Image Transformations," *OpenCV*. https://docs.opencv.org/master/d7/d1b/group__imgproc__misc.html (accessed Sep. 21, 2021).

[11] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015, Accessed: Sep. 21, 2021. [Online]. Available: http://arxiv.org/abs/1409.1556.

[12] "tf.keras.losses.CategoricalHinge," *TensorFlow*. https://www.tensorflow.org/api_docs/python/tf/keras/losses/CategoricalHinge (accessed Sep. 21, 2021).

[13] J. Weston and C. Watkins, "Support Vector Machines for Multi-Class Pattern Recognition," presented at the 7th European Symposium on Artificial Neural Networks, Bruges, Belgium, Jan. 1999.

[14] L. Poomhiran, P. Meesad, and S. Nuanmeesri, "Improving the Recognition Performance of Lip Reading Using the Concatenated Three Sequence Keyframe Image Technique," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 6986–6992, Apr. 2021, https://doi.org/10.48084/etasr.4102.

[15] S. Nuanmeesri, "Development of community tourism enhancement in emerging cities using gamification and adaptive tourism recommendation," *Journal of King Saud University - Computer and Information Sciences*, Apr. 2021, https://doi.org/10.1016/j.jksuci.2021.04.007.

[16] S. Nuanmeesri and W. Sriurai, "Thai Water Buffalo Disease Analysis with the Application of Feature Selection Technique and Multi-Layer Perceptron Neural Network," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 6907–6911, Apr. 2021, https://doi.org/10.48084/etasr.4049.

[17] A. N. Saeed, "A Machine Learning based Approach for Segmenting Retinal Nerve Images using Artificial Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 5986–5991, Aug. 2020, https://doi.org/10.48084/etasr.3666.

[18] H. Hasan, H. Z. M. Shafri, and M. Habshi, "A Comparison Between Support Vector Machine (SVM) and Convolutional Neural Network (CNN) Models For Hyperspectral Image Classification," *IOP Conference Series: Earth and Environmental Science*, vol. 357, Nov. 2019, Art. no. 012035, https://doi.org/10.1088/1755-1315/357/1/012035.

AUTHOR'S PROFILE

Sumitra Nuanmeesri received her Ph.D. in Information Technology at the King Mongkut's University of Technology North Bangkok, Thailand. She is an Assistant Professor in the Information Technology Department, Faculty of Science and Technology at Suan Sunandha Rajabhat University, Thailand. Her research interests include speech recognition, data mining, deep learning, image processing, mobile application, supply chain management system, internet of things, robotics, augmented reality, and virtual reality