

# Deep Metric Learning Based Citrus Disease Classification With Sparse Data

SIVASUBRAMANIAM JANARTHAN<sup>1,2</sup>, SELVARAJAH THUSEETHAN<sup>1</sup>,  
SUTHARSHAN RAJASEGARAR<sup>1,2</sup>, (Member, IEEE), QIANG LYU<sup>3,4</sup>, YONGQIANG ZHENG<sup>3,4</sup>,  
AND JOHN YEARWOOD<sup>1,2</sup>, (Member, IEEE)

<sup>1</sup>School of Information Technology, Deakin University, Geelong, VIC 3125, Australia

<sup>2</sup>Deakin-SWU Joint Research Centre on Big Data, Faculty of Science, Engineering and Built Environment, Deakin University, Geelong, VIC 3125, Australia

<sup>3</sup>Citrus Research Institute, Southwest University-Chinese Academy of Agricultural Sciences, Chongqing 400712, China

<sup>4</sup>Deakin-SWU Joint Research Centre on Big Data, Southwest University, Chongqing 400715, China

Corresponding author: Sutharshan Rajasegarar (srajas@deakin.edu.au)

**ABSTRACT** Early recognition of citrus diseases is important for preventing crop losses and employing timely disease control measures in farms. Employing machine learning-based approaches, such as deep learning for accurate detection of multiple citrus diseases is challenging due to the limited availability of labeled diseased samples. Further, a lightweight architecture with low computational complexity is required to perform citrus disease classification on resource-constrained devices, such as mobile phones. This enables practical utility of the architecture to perform effective monitoring of diseases by farmers using their own mobile devices in the farms. Hence, we propose a lightweight, fast, and accurate deep metric learning-based architecture for citrus disease detection from sparse data. In particular, we propose a patch-based classification network that comprises an embedding module, a cluster prototype module, and a simple neural network classifier, to detect the citrus diseases accurately. Evaluation of our proposed approach using publicly available citrus fruits and leaves dataset reveals its efficiency in accurately detecting the various diseases from leaf images. Further, the generalization capability of our approach is demonstrated using another dataset, namely the tea leaves dataset. Comparison analysis of our approach with existing state-of-the-art algorithms demonstrate its superiority in terms of detection accuracy (95.04%), the number of parameters required for tuning (less than 2.3 M), and the time efficiency in detecting the citrus diseases (less than 10 ms) using the trained model. Moreover, the ability to learn with fewer resources and without compromising accuracy empowers the practical utility of the proposed scheme on resource-constrained devices, such as mobile phones.

**INDEX TERMS** Citrus disease recognition, deep learning, metric learning, siamese network, sparse data.

## I. INTRODUCTION

The citrus fruit industry is one of the largest fruit industries in the world, with cultivation in 137 countries [1]. Continuous supply of citrus is vital for a healthy lifestyle of human beings due to its high Vitamin-C content and other useful nutrients. However, citrus crops are not only affected by various existing diseases but also constantly face the possibility of a devastating outbreak of newly emerging diseases around the world [2]. Although these diseases rarely last for long time periods, they cause a significant decrease in overall citrus production and create non-preventable economic depressions. Moreover, due to its rapidly dispersing nature, timely

recognition of the emergence of new diseases in citrus is vital to take effective disease control measures to safeguard crops. In particular, having the ability to detect the diseases directly in the farms using mobile devices will help to speed up this process. Hence, it is essential to develop a lightweight and fast framework that can accurately predict emerging diseases.

Several approaches for citrus disease recognition have been analysed in the past [3]–[8]. The traditional machine learning approaches, such as [4], [7], are time-efficient. However, a limitation in these approaches is the inability to achieve higher disease classification accuracies. They also heavily rely on domain expertise to extract hand-crafted features, which often contribute to a decline in the accuracy. The recent application of deep learning techniques, such as a deep convolutional neural network (DCNN) [3], [8], for citrus

The associate editor coordinating the review of this manuscript and approving it for publication was Wai Keung Fung <sup>1</sup>.

disease recognition has shown improved performances. It is, due to their ability to learn useful features from citrus samples automatically. However, these deep learning approaches require a significant amount of data (i.e., thousands of annotated samples) to obtain a generalised trained model. Citrus disease data collection is an expensive and time-consuming exercise due to the scale of citrus orchards and the varying nature of the disease dynamics across temporal and spatial scales. Due to the ever-changing symptoms of citrus diseases, it is also difficult to acquire sufficient infected samples that can be used for training machine learning models for automated disease detection. As a result, learning with sparse diseased data has become one of the major challenges for recent citrus disease recognition tasks. Recent works, such as [9], [10], and [3], proposed the use of generative adversarial networks to generate artificial diseased leaf samples as a potential solution for the sparse data problem. However, these approaches still struggle to achieve high detection accuracy rates.

Furthermore, the traditional deep learning models are not memory efficient. Consequently, these limit the applicability of these models on computers with limited memory and mobile devices. For example, deep neural networks, such as Inception-v3 [11], consume significant memory resources during run time. Some of the studies in the past proposed lightweight image classification techniques for use on devices with limited memory, such as mobile phones and tablet devices [12]–[15]. These lightweight models use effective computation techniques and less number of parameters for tuning, hence require low memory and less number of computations to complete a task. These properties enable those models to be deployed in resource constrained edge devices. For instance, in [16], a modified version of the MobileNet architecture has been proposed to perform plant disease classification. It uses a special technique called depthwise separable convolution, and only require 0.5 million parameters. However, it faces a trade-off between classification accuracy and memory consumption.

In order to address these challenges, inspired by [17], [18] and [19], we propose a novel deep metric learning-based framework with a patch generation mechanism to recognize citrus diseases from leaf images. Intuitively, humans can differentiate the infected and non-infected *local regions* of a leaf image to determine the disease. When a particular region of the leaf is identified as infected, the person may neglect the other regions of the leaf. Mimicking this behavior helps to formulate an efficient detection algorithm that can focus the attention on the region of the leaf where the most disease is appearing. Hence, in this work, we exploit this behavior by dividing the leaf into regions/patches and use for learning our proposed deep metric learning-based architecture. Our contributions in this work are summarized as follows:

- 1) We propose a novel *lightweight, fast, and accurate* deep metric learning-based citrus disease classification approach that can learn effectively with sparse data.

An *embedding module*, which is a deep convolutional neural network trained with a siamese objective, a *cluster prototype module*, which incorporates the K-Means clustering technique, and a *simple neural network classifier* are integrated to form a final classification network. We also developed a patch generation technique to improve the detection performance of our proposed framework.

- 2) Experimental results on the *citrus fruits and leaves dataset* [20] demonstrate that our proposed approach is fast, lightweight, and able to achieve higher classification accuracies.
- 3) Comparison evaluation with the state-of-the-art deep models, such as DenseNet-201 [21], Inception-v3 [11], VGG-16 [22], MobileNet [12], MobileNetV2 [13], NASNetMobile [14], and EfficientNetB0 [15] reveal that our proposed approach is capable of achieving high detection accuracy. Evaluation with an equivalent whole leaf-based model, and an ablation study using a patch-based learning demonstrate the improvements achieved in terms of detection accuracy due to the introduction of the various components and the patch creation method for our proposed architecture.

The rest of this article is arranged as follows. A review of existing citrus disease recognition approaches is given in Section II. Section III provides details of our proposed framework. The data pre-processing used in our experiments is described in Section IV. The experiments and results are reported with discussions in Section V, followed by conclusions and future work in Section VI.

## II. RELATED WORK

Automated citrus disease classification has been performed in the past using both handcrafted [4]–[7] and deep learning [3], [8] based feature extraction techniques. In this section, we review some of the closely related approaches in detail.

The majority of the handcrafted feature extraction techniques used in the literature are computationally less complex and have shown good state-of-the-art performances in various citrus disease classification tasks. For example, the hand-picked features from near-infrared spectra of the citrus canopy [4] have shown competitive accuracies in identifying the citrus greening disease. Sankaran and Ehsani [5] utilized fluorescence sensing to detect citrus greening disease, and obtained more than 94% accuracy in both laboratory and in the field conditions. Evidence of effectively identifying the citrus disease from fruits in addition to leaves is presented in [7]. To determine the canker disease from the citrus leaves, fruits, and canopy, in [6], different vegetation indices that are measured from the hyperspectral images are used. Their proposed method achieved better recognition accuracy for late-stage canker samples compared to early-stage canker samples. However, the main drawback of the handcrafted methods is the complexity in extracting and selecting pertinent features for different types of diseases. Thus, citrus

disease identification is limited to one or a few disease categories using the majority of handcrafted methods. Furthermore, the above methods utilized special sensing mechanisms, such as near-infrared spectra and fluorescence sensing, as opposed to regular images. This limits the ability to use these in practice in a handheld device, such as a mobile phone, which can take regular photos of leaves to process for diseases.

In contrast, deep learning-based automatic feature extraction techniques have performed well in both binary [3] and multi-class [8] citrus disease classification tasks. Zhang *et al.* [3] proposed a deep learning-based citrus disease classification model with a novel feature magnification and an optimization objective breakdown technique, which eliminates the overfitting issue that arises when using a small database. Pan *et al.* [8] proposed an alternative mobile-based citrus disease diagnosis system that avoids overfitting by employing various augmentation techniques and using a simplified densely connected convolutional network (DenseNet). Transfer learning is another relevant approach that demonstrated significant improvements in crops [23], apple [24], and cassava [25] disease classifications. However, for the citrus disease classification, greening is the only disease considered in [23]. Although the aforementioned early attempts showed some success, they still struggle from lower classification accuracies due to insufficient training samples. Hence in this work, we address the above limitations by proposing a joint framework using metric learning.

Deep metric learning-based techniques have been increasingly employed for a variety of classification tasks in the recent literature [17]–[19], [26]. Koch *et al.* [17] proposed a convolutional siamese network for one-shot learning. Snell *et al.* [18] proposed a prototypical network for few-shot learning, where a single prototype obtained from a support set is used to verify a query set. Contrarily, Rippel *et al.* [19] proposed a magnet loss function to train the model considering both interclass and intraclass variations in the sample space. A soft k-nearest-cluster metric-based evaluation is used to perform the image classification task. Inspired by this, in our work, we employ three well-utilized techniques, namely deep siamese networks, K-Means clustering, and neural network classifiers to improve the disease detection performance. Next, we present our proposed framework in detail.

### III. PROPOSED MODEL

In this section, the overall architecture of our proposed deep metric learning-based citrus disease classification framework is presented. The citrus leaf samples are first pre-processed to remove the background, create the patches, and augment the images as discussed in Section IV. These patches are then used in our framework for disease classification. Below are the steps involved in this process.

**Step 1:** During the pre-processing step (see Section IV) each whole leaf is taken and divided into five patches. These leaf patches are then further divided into

eligible patches (those that have clear symptoms of disease) and non-eligible patches (those that do not have clear symptoms of disease) as explained in Section IV-C. A collection of similar and dissimilar pair pools are formed from the eligible patches. Note that a pair of leaf patches are called similar if they belong to the same disease. Then a pair of patches  $P_1$  and  $P_2$  are selected from the similar or dissimilar pair pools and presented to a trainable DCNN embedding module  $G_W$  with shared weights  $W$ . The output of this module,  $G_W(P_1)$  and  $G_W(P_2)$ , represent the corresponding embeddings of the input patches. An Euclidean neuron  $D_W$  computes the distance between these embeddings  $G_W(P_1)$  and  $G_W(P_2)$ . Then, the contrastive loss  $L_C$ , which is the difference between the computed distance and the ground truth, is used with back propagation to train the deep embedding module  $G_W$ . This step is further explained in subsection III-A. The trained embedding module  $G_W$  is used in the subsequent steps 2 and 3.

**Step 2:** The embeddings are calculated for all the eligible patches using the pre-trained embedding module  $G_W$ . A K-Means clustering algorithm is applied to cluster the computed embeddings, and the cluster prototypes obtained, are  $\{C_1, C_2, \dots, C_{20}\}$ . These cluster prototypes are further used in Step 3. More details of this process are discussed in subsection III-B.

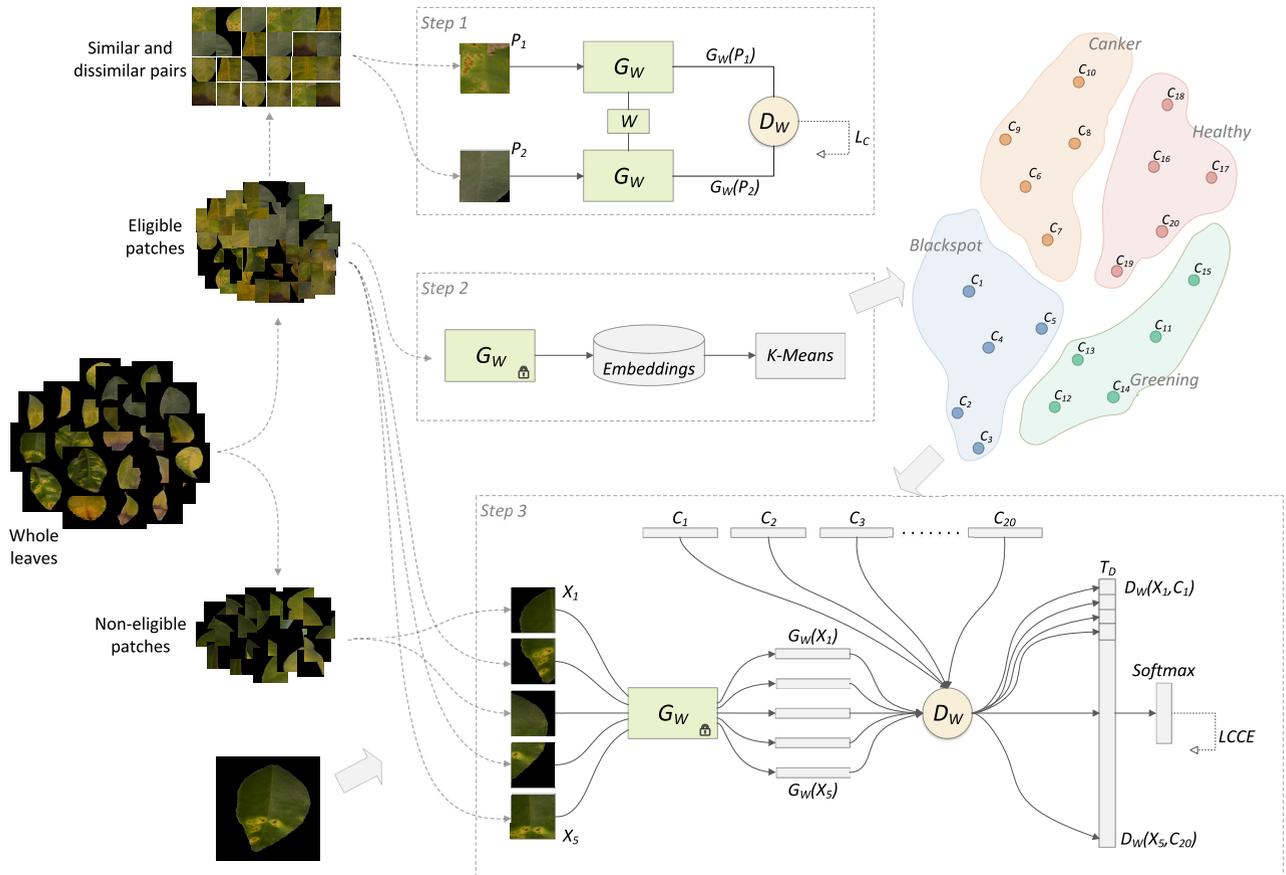
**Step 3:** The patch embeddings  $\{G_W(X_1), G_W(X_2), \dots, G_W(X_5)\}$  for the five patches  $\{X_1, X_2, \dots, X_5\}$  of a leaf are computed with the pre-trained embedding module  $G_W$ . Subsequently, using the Euclidean distance neuron  $D_W$ , the distances  $\{D_W(X_1, C_1), D_W(X_1, C_2), \dots, D_W(X_5, C_{20})\}$  between each pair of patch embeddings and the cluster prototypes are computed to form a distance tensor  $T_D$ . Finally, the distance tensor  $T_D$  is presented to the *Softmax* layer and trained with categorical cross-entropy loss  $L_{CCE}$ . Further details about this process are discussed in subsection III-C.

The execution of the above steps results in a trained citrus disease classification network. During the testing process, the leaf images are presented to the learned classifier network (shown in Step 3 of Figure 1) to classify them as either belonging to the normal (non-diseased) class or one of the disease classes.

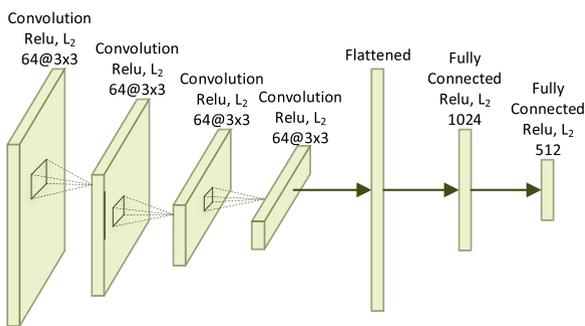
Next, we discuss each of the components of our framework in detail.

#### A. DEEP SIAMESE NETWORK

A siamese network provides a unique structure to score similarity between inputs [17], [27]. In [27], the siamese concept was introduced to compare the handwritten signatures using two parallel neural network components. This concept has



**FIGURE 1.** Citrus disease classification framework.  $P_1, P_2$ : patches belong to a similar/dissimilar pair,  $G_W$ : DCNN embedding module as shown in Figure 2,  $W$ : shared weight,  $D_W$ : Euclidean distance neuron,  $L_C$ : contrastive loss, Embeddings: the embeddings of eligible patches, K-Means: K-Means clustering algorithm,  $C_1 - C_{20}$ : cluster prototypes,  $X_1 - X_5$ : patches of a leaf,  $T_D$ : euclidean distances between the embeddings of input patches and cluster prototypes,  $Softmax$ : softmax output layer,  $L_{CCE}$ : categorical cross entropy loss.



**FIGURE 2.** The deep convolutional neural network (embedding module  $G_W$  as shown in Figure 1).

been extended in [17] using a DCNN component instead of the traditional neural network for the siamese network, which demonstrated good performances in diverse image classification tasks. During the siamese network training, embeddings of similar and dissimilar patches are learnt. These learnt embeddings are eventually used to calculate the distances between the two samples of a pair, followed by using them for calculating the loss against the ground truth, as described

in III-A2. In order to obtain the embedding of these patches, we formulated a carefully crafted DCNN based deep siamese network, as shown in Step 1 of Figure 1. The internal components of the DCNN framework  $G_W$ , namely the embedding module, is shown in Figure 2. We constructed various layers of the DCNN component using the methods presented in [18], [28].

For training and validation of the deep siamese network, a *sample selection* procedure that generates batches of images consisting of equal numbers of similar and dissimilar samples is used. Further, a *contrastive loss function* ( $L_C$ ) is used as the objective function to optimize our model. Next, we explain these in more detail.

### 1) SAMPLE SELECTION

First, all the permutations for each citrus leaf samples,  $P_T$ , are created. Second, the sample pairs are separated into a similar pair pool,  $P_S$ , and a dissimilar pair pool,  $P_D$ . A pair is considered *similar* if both the samples of a pair belong to the same class (disease), and dissimilar otherwise. Let  $C$  be the number of classes, and assume that each class contains  $N$  samples. Then, the number of pairs  $P_T$ ,  $P_S$ , and  $P_D$  can be

defined as shown in Eq. 1, 2, and 3, respectively.

$$P_T = \frac{CN(CN - 1)}{2} \quad (1)$$

$$P_S = \frac{CN(N - 1)}{2} \quad (2)$$

$$P_D = P_T - P_S = \frac{C(C - 1)N^2}{2} \quad (3)$$

Further, the ratio between similar and dissimilar pairs,  $r_{sd}$ , based on  $P_S$  and  $P_D$  is defined as follows:

$$r_{sd} = \frac{P_S}{P_D} = \frac{CN(N - 1)}{CN^2(C - 1)} = \frac{N - 1}{N(C - 1)} \approx \frac{1}{C - 1} \quad (4)$$

where  $N$  and  $N - 1$  are approximately equal for large datasets.

As can be seen in Eq. 4,  $P_D$  is approximately  $C - 1$  times higher than  $P_S$ . This makes the training data highly imbalanced, which causes the model to bias towards  $P_D$ . Hence, in order to eliminate this issue, a random selection mechanism is used. For every batch in the training sample of size  $B$ , pairs of size  $B/2$  are randomly selected from  $P_S$ , while the rest of the  $B/2$  pairs are selected from  $P_D$ . Also, when each pair is drawn from the pool, for each sample of the pair, a randomly selected augmentation is applied.

## 2) CONTRASTIVE LOSS FUNCTION

We use the contrastive loss function introduced in [29], which is a pair based loss function that defines the mapping of similar samples to nearby points, and dissimilar samples to distant points in the output manifold, for optimisation. In order to estimate the contrastive loss, the distance  $D_W$  between the two outputs of  $G_W$  for a pair of input samples  $P_1$  and  $P_2$  respectively is computed, as shown in Eq 5.

$$D_W(P_1, P_2) = \|G_W(P_1) - G_W(P_2)\|_2 \quad (5)$$

The contrastive loss,  $L_C$ , with margin  $m$  is then estimated as follows:

$$L_C = (1 - Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{\max(0, m - D_W)\}^2 \quad (6)$$

where  $m > 0$ . Note that the dissimilar pairs contribute to the contrastive loss only when  $D_W < m$ .  $Y$  is the label assigned to an input pair as per the conditions defined below.

$$Y = \begin{cases} 0 & \text{if } P_1 \text{ and } P_2 \text{ belong to the same class} \\ 1 & \text{if } P_1 \text{ and } P_2 \text{ belong to different classes} \end{cases} \quad (7)$$

The validation accuracy and loss will not reflect the best values since the training and validation of the deep siamese network are based on random input pairs. Hence, in order to select the best model, all the models with the validation accuracy greater than a threshold (we choose 85% validation accuracy) are saved. The saved models are then re-evaluated with the entire validation set, and the model with the highest validation accuracy is selected. Finally, the embedding module, loaded with the weights of the best model, is used in Step 2 to compute the embeddings as well as in Step 3 for the final classification task.

## B. CLUSTER PROTOTYPES

During step 2 (in Figure 1) of the citrus disease classification framework, the eligible patches are grouped into distinct clusters, and the cluster prototypes (the center of each cluster)  $\{C_1, C_2, \dots, C_{20}\}$  are obtained and eventually used for training the classifier network (in Step 3 of Figure 1).

As illustrated in Step 2 of Figure 1, the embeddings are first computed for eligible patches using the pre-trained embedding module  $G_W$ . Next, the embeddings are grouped into clusters using a K-Means clustering algorithm, where the number of clusters  $k$  is set to 5 by considering the intraclass variations of the diseased citrus leaf samples. Therefore, for the 4 disease classes, a total of 20 prototypes are obtained. These created prototypes are used in Step 3, which is our final classification network.

## C. CLASSIFICATION NETWORK

As shown in Figure 1, classification network is the major component of our proposed approach, which comprises the following key components, namely, 5 input patches of a leaf image  $\{X_1, X_2, \dots, X_5\}$ , the pre-trained embedding module  $G_W$  obtained from Step 1 (as shown in Figure 2), cluster prototypes obtained from Step 2  $\{C_1, C_2, \dots, C_{20}\}$ , a neuron  $D_W$  that calculates the Euclidean distance, a *Softmax* output layer, and a categorical cross-entropy loss  $L_{CCE}$  component.

First, the embeddings  $\{G_W(X_1), G_W(X_2), \dots, G_W(X_5)\}$  are computed using the embedding module  $G_W$  for all the five patches of a leaf. Second, the distances  $\{D_W(X_1, C_1), D_W(X_1, C_2), \dots, D_W(X_5, C_{20})\}$  are calculated between each pair of the embeddings  $\{G_W(X_1), G_W(X_2), \dots, G_W(X_5)\}$  and the cluster prototypes  $\{C_1, C_2, \dots, C_{20}\}$  to form a distance tensor  $T_D$ . Finally, the distance tensor  $T_D$  is given as an input to the simple neural network classifier with a *Softmax* layer. During the training process, along with the input distance tensor  $T_D$ , the categorical cross entropy loss  $L_{CCE}$  is used to train the *Softmax* layer.

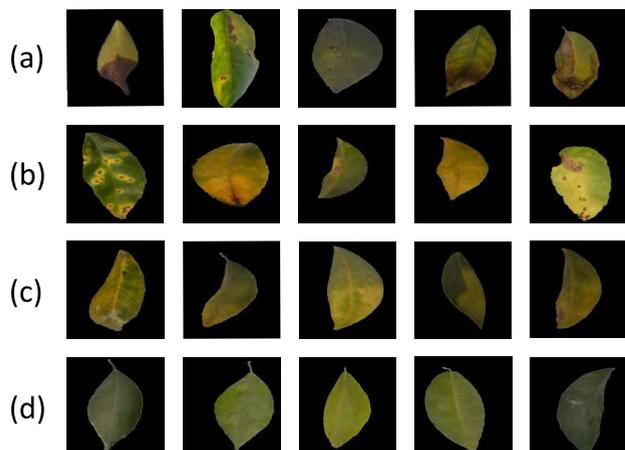
## IV. DATA PRE-PROCESSING

In this section, we discuss the data pre-processing performed on the citrus leaf images before used in the proposed classification framework. It comprises several steps, namely background removal, patch creation and labelling and image augmentation. We first describe the publicly available citrus fruits and leaves dataset [20] that we use for our evaluation, followed by the various steps of the pre-processing procedure using this data.

### A. DATASET

In this study, the experiments are performed using a publicly available citrus fruits and leaves dataset [20]. The dataset contains a total of 609 annotated leaf images categorised into healthy and four citrus disease classes, namely *black spot*, *canker*, *greening*, and *melanose*. The original dataset was created by taking the photos of the fruits and leaf samples collected from Sargodha region in Pakistan. Authors have

used a Canon EOS 1300D advanced DSLR camera with  $5202 \times 3465$  resolution to acquire the images. The collected images were resized to  $256 \times 256$  size with 72 dpi resolution [20]. The published dataset contains images with 256 shades for each RGB layer in sizes of  $256 \times 256$  pixels.



**FIGURE 3.** Citrus diseased and healthy leaf samples (a) Black spot (b) Canker (c) Greening (d) Healthy.

Note that, in this study, the samples belonging to *melanose* disease were eliminated since only a few samples of that disease category were present in the dataset (8 leaf samples) compared to other classes. Figure 3 shows the sample images for each of the four selected disease classes. Further, the defect samples, such as the same leaf images that appear in more than one disease class, and the (exact) duplicate samples found within the same disease class were removed from the original database. The above data cleaning process resulted in a reduction of around 18.95% (i. e., 596 to 483) of the samples, as presented in Table 1; see the second and third columns *Initial leaf count* and *Final leaf count*. Further, the intra-class variations can be observed from the images from the same class/disease in Figure 3.

**TABLE 1.** Counts of leaves and patches found in the diseased citrus leaves.

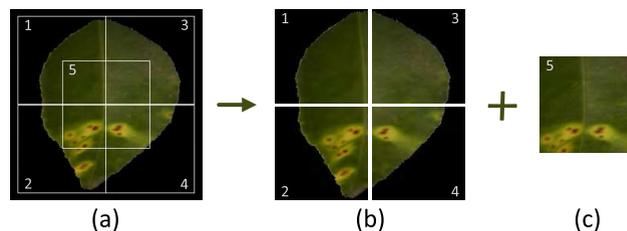
Disease	Initial leaf count	Final leaf count	Total patch count	Eligible patch count
Black spot	171	124	620	430
Canker	163	157	785	692
Greening	204	144	720	647
Healthy	58	58	290	288
Total	596	483	2415	2057

The citrus fruits and leaves dataset is considered as sparse data since it contains only a small number of samples for each disease class, i.e., a maximum of 157 samples per class. This presents a challenge in learning an accurate model using some of the the state-of-the-art deep networks, such as DenseNet-201 [21], Inception-v3 [11], and VGG-16 [22], which require thousands of images for training purposes.

During the pre-processing step, the background of each leaf image is removed and the patches (regions) are created, as we explain next.

**B. BACKGROUND REMOVAL**

DCNN is capable of learning the foreground features in the presence of background in an image. However, the feature complexity can be reduced by removing the background and in turn, helps improve the model accuracy [30]. Hence, we perform the background removal of leaf images using a two-step image segmentation approach. In the first step, a segmentation algorithm,<sup>1</sup> which utilises a K-Means clustering technique followed by an active contour refinement, is used for creating the leaf masks. The extracted masks are then used to automatically remove the backgrounds of the leaf images, which resulted in 99% of the images being successfully processed. In the second step, for the failed images (around 1% of them) resulted from the above automated background segmentation process, a manual segmentation is performed. The final segmented leaf images are then used to create the patches.



**FIGURE 4.** Patch generation process (a) whole leaf of canker disease (b) top-left, top-right, bottom-left and bottom-right patches and (c) center patch.

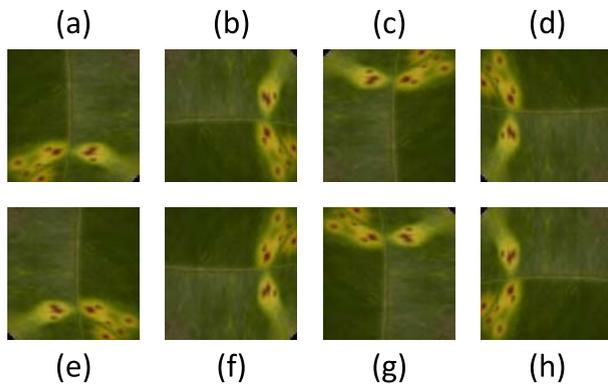
**C. PATCH CREATION AND LABELING**

In general, citrus disease symptoms usually occupy only a small area or region of the whole leaf (e.g., see Figure 4). Hence, rather than using the whole leaf image for processing, it will be advantageous to divide the whole leaf into small regions (patches) and use for training a classifier. This provides more focused image patches to learn the intrinsic features of the symptoms of a disease accurately by the deep network, enabling higher detection accuracy. Hence, we create multiple patches from each image and relabeled them appropriately. To perform this task, each leaf image is split into five patches, namely top-left, top-right, bottom-left, bottom-right, and center, as illustrated in Figure 4.

Note that, intuitively, a human can differentiate the infected and non-infected local regions in a leaf image to help judge the disease. When a particular region of the leaf is identified as infected, humans may neglect the other regions of the leaf. This is a handy behaviour that can be mimicked to select the patches for training to improve the trained model, and hence we adopted this approach in our framework to

<sup>1</sup><https://au.mathworks.com/help/images/image-segmentation-using-the-image-segmenter-app.html>

create patches. The obtained patches are then categorized into eligible or non-eligible patches based on the disease symptoms and size of the leaf area they occupy. For example, top-left and top-right patches from the leaf shown in Figure 4 are marked as non-eligible patches since they do not exhibit any noticeable symptoms of the disease. The patches with a small leaf area, i. e., the majority of the patch containing background information, are also marked as non-eligible patches. Only the eligible patches are used in Step 1 and Step 2 of our framework as shown in Figure 1, while both eligible and non-eligible patches are used in Step 3. In the context of healthy images, size of the leaf area within the image is used as the only criteria to select the eligible patches. In Table 1, the columns *Total patch count* and *Eligible patch count* indicate the number of created patches and the eligible patches for each disease category, respectively.



**FIGURE 5.** Illustration of the samples created through augmentations. (a) original (b) rotated  $90^\circ$  (c) rotated  $180^\circ$  (d) rotated  $270^\circ$  (e) horizontal flip (f) horizontal flip and rotated  $90^\circ$  (g) vertical flip (h) vertical flip and rotated  $90^\circ$ .

#### D. AUGMENTATION

In the augmentation step, we employed rotation (counter-clockwise) and flipping (vertical and horizontal) for the image samples to further increase the size of the training set. The original image is rotated by  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$  angles to generate four additional augmented samples. Further, the vertical and horizontal flips are applied to the original image followed by a  $90^\circ$  rotation of the flipped samples to form another four augmented images. Accordingly, altogether eight unique augmented samples are generated from each image sample (patch), as shown in Figure 5.

#### V. EVALUATION

This section presents the results obtained on the citrus fruits and leaves dataset [20] for our models. The aim of the evaluation is to assess the proposed patch-based citrus leaf classification framework's ability to learn from sparse samples effectively as well as classifying the diseased samples accurately. We use classification accuracy as the evaluation metric, and the results are reported after 5-fold cross-validation. We further report the learning and validation curves along with the confusion matrices to elaborate the outcomes graphically.

#### A. IMPLEMENTATION AND TRAINING

In all experiments, the models were implemented using TensorFlow<sup>2</sup> framework and accelerated on NVIDIA GeForce GTX 1080 Titan GPU. For training, both variants of our model, namely Our Model-128 and Our Model-112, having input images of sizes  $128 \times 128 \times 3$  and  $112 \times 112 \times 3$  were used along with an Adam [31] stochastic optimizer, respectively. For the siamese network, the batch-size is set to 32, the number of steps is set to 250, and trained for 200 epochs with a learning rate of 0.00006. We changed the batch-size to 8 and trained for 50 epochs with an updated learning rate of 0.001 for the classification network. The training accuracy and loss changes obtained for both the siamese and the classification networks are illustrated in Figure 6.

For training the other state-of-the-art models, we trained all the layers after initializing them with the ImageNet weight with a set of the same hyperparameters as used in [8]. Stochastic gradient descent (SGD) optimization is used to train the networks with the momentum set to 0.9. The initial learning rate was set to 0.001 and multiplied by 0.94 after every two epochs. The input image size is set to  $256 \times 256 \times 3$  for all the models, except for the MobileNet, MobileNetV2 and NASNetMobile. For the MobileNet, MobileNetV2 and NASNetMobile, the input image size is set to  $224 \times 224 \times 3$ , as it is the biggest image size variant available with ImageNet weight. Finally, the batch size is set to 8 and trained the MobileNetV2, NASNetMobile and EfficientNetB0 up to 200, 1000, and 100 epochs respectively, and all the other networks are trained up to 50 epochs until they converge.

On the other hand, when using the tea leaf dataset [32], the siamese network is trained for 200 epochs after changing the batch size to 8 and the number of batches to 50. For all the other cases, the batch size is changed to 4 and all the other parameters are kept unchanged.

We trained two variants of our proposed model to validate the effectiveness of our framework against the state-of-the-art models. Our Model-128 is trained with the input image size of  $128 \times 128$  for a fair comparison with the state-of-the-art models, such as DenseNet-201, Inception-v3, VGG-16, and EfficientNetB0, which are trained with an image size of  $256 \times 256$ . On the other hand, Our Model-112 is trained with  $112 \times 112$  size images for a fair comparison with the other state-of-the-art models, such as MobileNet, MobileNetV2, and NASNetMobile, which are trained with  $224 \times 224$  size images. In both cases, since our models are trained with five patches of each leaf, they have processed 81,920 pixels ( $128 \times 128 \times 5$ ) and 62,720 pixels ( $112 \times 112 \times 5$ ), respectively. In contrast, the state-of-the-art models processed 65,536 pixels ( $256 \times 256$ ) and 50,176 pixels ( $224 \times 224$ ), respectively. Even though our models process higher number of pixels (25% more in both cases), they show comparable time efficiency against the state-of-the-art lightweight models, which is further discussed in the next section.

<sup>2</sup><https://www.tensorflow.org/>

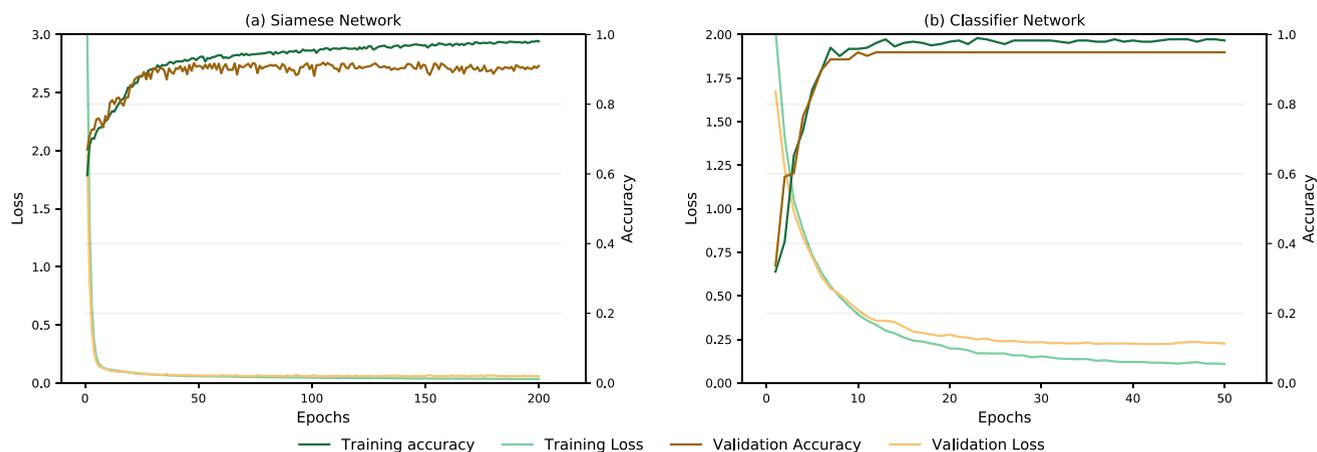


FIGURE 6. Accuracy and loss curves for the siamese (left) and classification (right) networks.

## B. RESULTS AND DISCUSSION

Overfitting is one of the main challenges faced in training a deep learning network with sparse data. The siamese network and the classification network of our proposed framework are able to overcome this challenge as demonstrated in Figure 6. To achieve this, in the siamese network, the dataset is expanded by creating patches. In addition, since a pair based approach is utilized for the training of the siamese network, the constructed pair pools generated significantly large sample set. On the other hand, for the training of the classifier network, the sample size used is still small. Hence, in order to avoid the overfitting problem, a simplified network with only one layer is used. Further, we shuffled the input patches  $\{X_1, X_2, \dots, X_5\}$  each time when a leaf sample is drawn for training. In this process, each leaf will take one of the  $5!$  input positions, and therefore resulting in the distance tensor  $T_D$  taking one of the  $5!$  forms for the same leaf sample, in each draw. This process is equivalent to increasing the number of training samples by  $5!$  (120) times, and hence helps to improve the generalization capability of the final classification network.

Figure 7 shows the confusion matrices obtained for the classification of each disease. As can be seen from the results, our models performed well for canker, black spot, and healthy classes, achieving greater than 94% classification accuracy. However, it can be observed that the lowest accuracy is reported for the greening disease since some of the samples from the greening disease have evidence of black spot disease. This can be clearly seen from the misclassified image samples (greening and black spot) shown in Figure 8. In Figure 8 (a), the leaf shown is genuinely affected by the black spot disease, which is given as a reference. In contrast, Figure 8 (b)-(d) are the misclassified leaf samples, where the symptoms of black spot disease can be observed to a certain degree. Not surprisingly, the major portion of the failed samples from the black spot class were also classified as greening. This behavior is because the black spot symptoms tend to

blend with the greening disease symptoms when the severity of the black spot disease is not intense, as can be observed in Figure 8. Further, It can be observed that both of our models achieved similar accuracies for all the classes even though the input image size is changed. The change in the input image size has only reduced the image width and height by 12.5%. This reveals that the change in this level of image size has not impacted heavily the features extracted by the convolution layers of the siamese network for measuring the similarity score, and hence the classification accuracy. However, the reduction in the image input size helps reduce the number of parameters of the network, and hence the complexity of the framework.

To further analyze the classification pattern of our proposed architecture, we generated a t-SNE [33] representation of the embeddings (obtained in Step 2) and the distances between the cluster prototypes (obtained in Step 2), which are shown in Figure 9. As can be seen in Figure 9 (a), the healthy and canker classes are clearly separated from the other classes with less misplaced embeddings. However, the embeddings of the black spot and greening classes are highly overlapped, and show a considerable amount of misplaced embeddings. We can relate this with our confusion matrix outcome shown in Figure 7. The greening and black spot diseases show a high level of misclassification between each other, with 6.95%, 6.25% (from greening to black spot) and 4.03%, 3.22% (from black spot to greening) misclassification for both of our models, respectively. The distances between the class prototypes are presented in Figure 9 (b), which shows the closeness of the embeddings in each disease class. From the figure, it can be inferred that the distances between the cluster prototypes for healthy are smaller (close to each other) compared to the other classes. Further, we can observe significant intraclass variations within each class of the diseased leaf samples, and this can also be observed in Figure 3. Similarly, we can observe around 5 intraclass variations for black spot and canker diseases. This provides support for the selection of

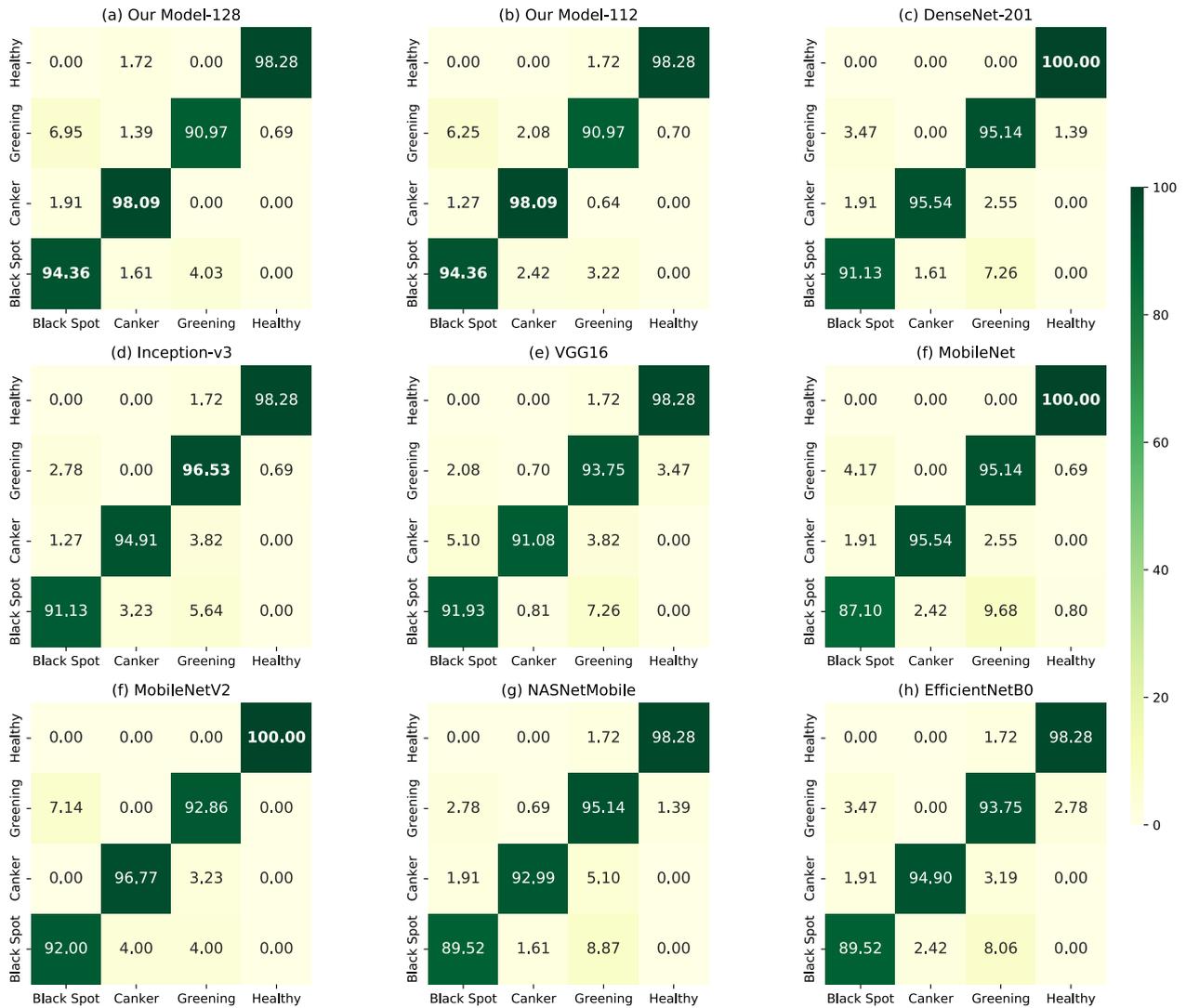


FIGURE 7. Confusion matrices for both of our models and the other state-of-the-art models obtained using citrus fruits and leaves dataset [20].

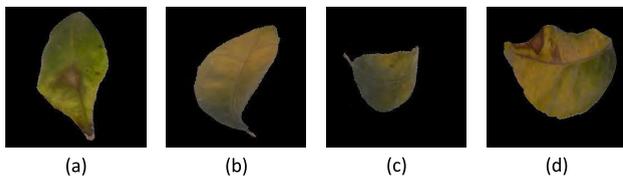


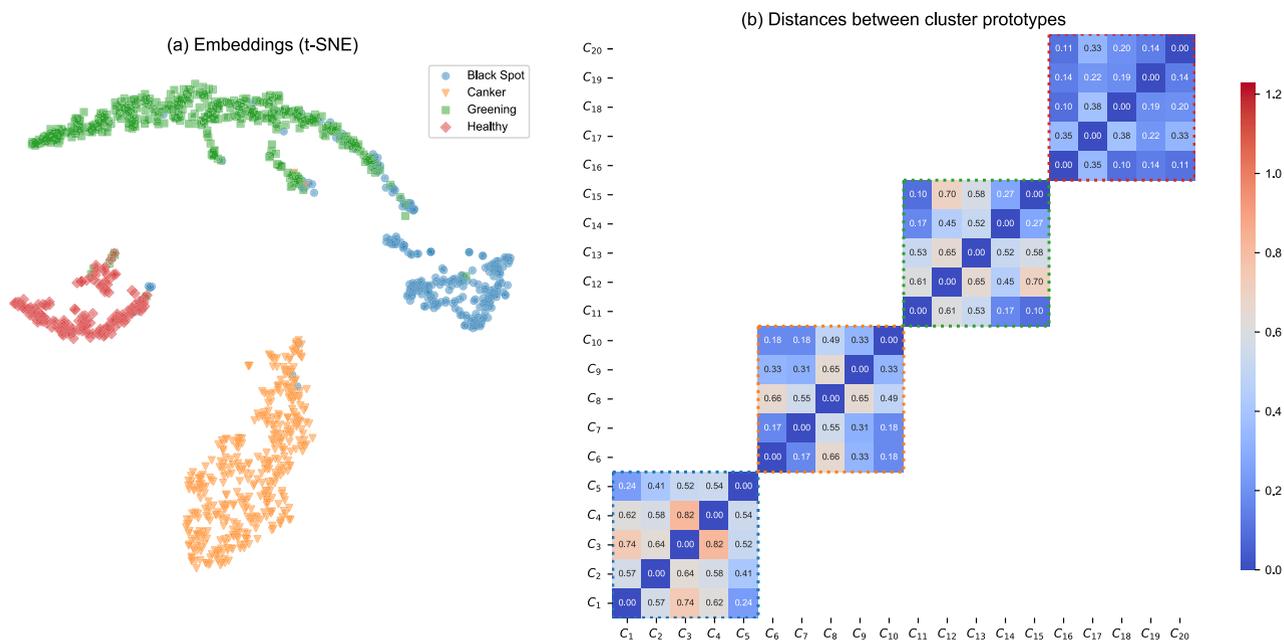
FIGURE 8. Examples of misclassified samples. (a) reference image for black spot disease, (b)-(d) misclassified samples that have symptoms of black spot disease in addition to the greening disease.

our  $k$  value (number of cluster prototypes) of the K-Means clustering algorithm to be 5, uniformly across all the classes. However, we believe that selecting the number of cluster prototypes empirically as well as using different  $k$  values for each class (disease) may help to improve the accuracy further, and we leave this analysis for future work.

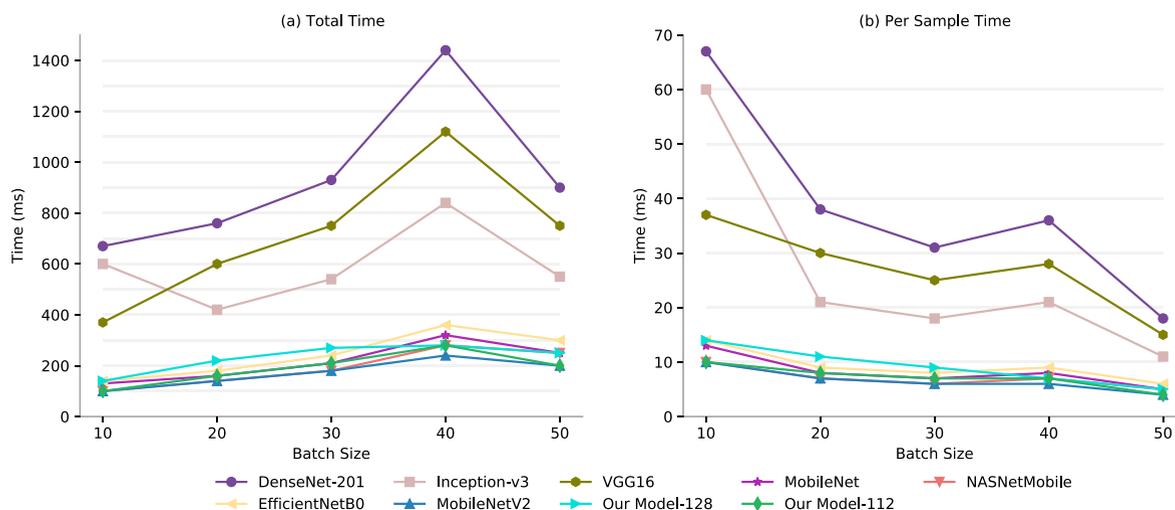
We have also compared our models with existing state-of-the-art deep networks in terms of classification accuracy and time efficiency. The total testing time and per sample

testing time for both our models and the state-of-the-art deep models for various batch sizes are shown in Figure 10. From the figure, it can be observed that the heavier models, namely DenseNet-201 [21], VGG16 [22], and Inception-v3 [11] take more time, and the MobileNet [12], MobileNetV2 [13], NASNetMobile [14], and EfficientNetB0 [15], which are the lightweight models aimed for use with the resource constrained devices, take lesser time to predict the diseases. Further, by comparing the time taken by both variants of our models, it can be inferred that the proposed models show comparable time efficiency with the state-of-the-art lightweight models. Overall, the time behavior of the proposed models are similar to MobileNet [12], MobileNetV2 [12], NASNetMobile [14] and EfficientNetB0 [15]. Further, they show similar per sample time for different test batch sizes. This behaviour is advantageous, especially when it comes to predicting a larger number of disease samples.

We further compare the results between both variations of our models against the seven state-of-the-art deep networks in



**FIGURE 9.** (a) Illustration of the embeddings (obtained in Step 2) using t-SNE diagram. (b) Distances between cluster prototypes (obtained in Step 2) in each class (disease). C<sub>1</sub> – C<sub>5</sub> - Black Spot, C<sub>6</sub> – C<sub>10</sub> - Canker, C<sub>11</sub> – C<sub>15</sub> - Greening, C<sub>16</sub> – C<sub>20</sub> - Healthy.



**FIGURE 10.** Comparison of testing time between our models and the other deep networks. (a) Total time taken to predict different batch sizes (b) Per sample time taken to predict different batch sizes.

Table 2. Both of our models achieve 95.04% accuracy. Compared to VGG16 [22], NASNetMobile [14], EfficientNetB0 [15] and MobileNet [12], our models show better classification accuracies by a clear margin. Compared to MobileNetV2 [13], Inception-v3 [11], and DenseNet-201 [21], our models show a slightly better overall classification accuracies. In particular, as can be seen from Figure 7, our models are superior to other deep networks for canker disease classification. We can also observe from the precision, recall, and F1 scores, reported in the Table 2, that both our and the state-of-the-art models are not impacted by the data imbalance, which is caused by the significantly lower number of healthy samples present in the citrus disease dataset. This may be due to the fact that the healthy samples are easier to learn as they

are very similar and do not have complex features to learn and generalize. Note that in terms of resource requirements, next to MobileNetV2 ( $\approx 2.26$  million), our models require a fewer parameters ( $\approx 2.99$  and  $\approx 2.27$  million) as opposed to the other networks ( $\approx 3.23$  million for MobileNet,  $\approx 4.05$  million for EfficientNetB0,  $\approx 4.27$  million for NASNetMobile,  $\approx 18.32$  million for DenseNet-201,  $\approx 14.71$  million for VGG16, and  $\approx 21.81$  million for Inception-v3). This property of our models demonstrates their ability to classify the diseases in a resource-constrained environment, such as on a mobile phone.

In order to show the generalization ability of our models, we compared the recognition accuracy of our models against existing benchmark deep learning models on another dataset,

**TABLE 2.** 5-fold cross-validation accuracy, precision, recall, and F1 Score comparison using citrus fruits and leaves (Citrus) [20] dataset, and 5-fold cross-validation accuracy comparison using tea leaves (Tea) [32] dataset.

Method	Input shape	Citrus				Tea	Number of parameters
		Precision	Recall	F1 Score	Accuracy(%)	Accuracy(%)	
Inception-v3 [11]	256 × 256 × 3	<b>0.9549</b>	0.9523	0.9524	94.79	99.16	≈ 21.81 M
DenseNet-201 [21]	256 × 256 × 3	0.9507	0.9544	0.9519	94.80	<b>100.00</b>	≈ 18.32 M
VGG16 [22]	256 × 256 × 3	0.9316	0.9377	0.9319	92.93	97.49	≈ 14.71 M
MobileNet [12]	224 × 224 × 3	0.9432	0.9442	0.9422	93.78	98.33	≈ 3.23 M
MobileNetV2 [13]	224 × 224 × 3	0.9489	0.9510	0.9487	94.37	99.16	≈ <b>2.26 M</b>
NASNetMobile [14]	224 × 224 × 3	0.9408	0.9395	0.9388	93.35	98.33	≈ 4.27 M
EfficientNetB0 [15]	256 × 256 × 3	0.9366	0.9412	0.9376	93.55	98.33	≈ 4.05 M
Our Model-112	112 × 112 × 3	0.9546	<b>0.9547</b>	<b>0.9541</b>	<b>95.04</b>	98.33	≈ 2.27 M
Our Model-128	128 × 128 × 3	0.9547	0.9546	0.9538	<b>95.04</b>	99.16	≈ 2.99 M

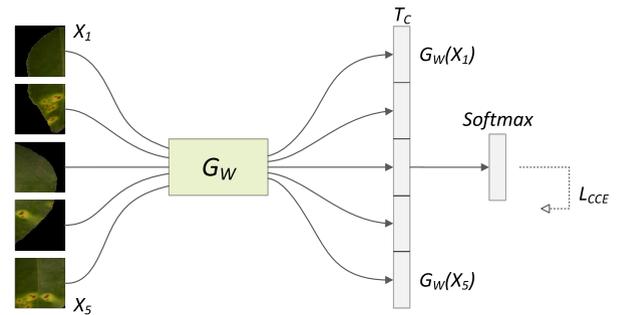
**TABLE 3.** Counts of leaves and patches of diseased tea leaves.

Disease	Leaf count	Total patch count	Eligible patch count
Leaf Blight	40	200	161
Red Scab	40	200	187
Red Leaf Spot	40	200	177
Total	120	600	525

namely the tea leaves data set [32]. It contains 40 images for each disease, such as leaf blight, red scab, and red leaf spot as shown in Table 3. The results are promising as illustrated in Table 2. Our Model-128 clearly outperformed VGG16 [22], and shown comparable performance against other state-of-the-art networks, which are trained with image size of 256 × 256. Further, Our Model-112 has shown comparable performances with other models that are trained with 224 × 224 size images, albeit with lower or comparable computational overhead.

1) WHOLE LEAF-BASED MODEL

We also built a variant of our proposed model to use the whole leaf as an input instead of the patches, and performed the experiments to compare against our proposed patch-based models. In order to build the whole leaf-based model, in Step 1, both similar and dissimilar pairs of whole leaves are used to train the siamese network. Similarly, in Step 2, the embeddings are computed for all the whole leaves in the training set and clustered them to obtain 20 cluster prototypes. Finally, in Step 3, the whole leaf images are fed, instead of the patches (5 patches per leaf), to the final classification network. Further, instead of using the DCNN, as shown in Figure 2, we used the one proposed in [17], as it demonstrated good performances during the siamese training for the whole leaf. Further, we performed 5-fold cross-validation evaluation using the citrus leaves dataset [20].



**FIGURE 11.** Fused CNN network used for ablation study,  $G_W$ : DCNN embedding module (trainable) as shown in Figure 2,  $T_C$ : concatenated embedding tensor,  $Softmax$ : softmax output layer,  $L_{CCE}$ : Categorical cross entropy loss.

In the evaluation, the whole leaf-based model achieved 90.28% accuracy, which is 4.76% lower than our proposed patch-based models. A possible reason for the lower accuracy is due to the reduction in the number of sample pairs (whole leaves) used for training the siamese network, compared to the patch-based training method, leading to an under generalized learnt siamese network model. In addition, since the whole leaf was used for the training, the healthy portion (region) of a diseased leaf may have introduced noise during the similarity computation process of the siamese network. However, for the patch-based scheme, only the patches that have shown clear disease symptoms (eligible patches) are used for training.

2) ABLATION STUDY

We performed an ablation study to compare our deep metric learning-based method against an equivalent traditional CNN based architecture. In order to perform that, a fused CNN network is built using the same embedding module  $G_W$ , as shown in Figure 11. All the 5 patches of each leaf are presented to the DCNN embedding module, and then the resulting embeddings are concatenated and presented to the final Softmax output layer. Five-fold cross-validation is performed using the citrus leaves dataset [20] with an input image size of 128 × 128, which achieved a 92.13% accuracy. This accuracy is 2.91% lower than that of our proposed models. Hence, the results demonstrate that our proposed patch-based framework significantly improves the accuracy in detecting the diseases in citrus leaves.

## VI. CONCLUSION AND FUTURE WORK

This article presents a deep metric learning-based framework to recognize citrus diseases effectively from leaf images. The proposed architecture comprises an embedding module, a cluster prototype module, and a simple neural network classifier to perform the disease recognition. An approach to generate patches from the leaf images is also included in the framework to further enhance the performance. Comparison evaluation with the whole leaf-based model and an ablation study demonstrated the improved performances achieved when our metric learning based architecture is combined with the patch generation mechanism. Comparison of our method with other deep network baselines in terms of time efficiency showed comparable or superior performances with other baselines. Further, our framework has shown better classification accuracy than all the other baselines. Our experiment with the tea leaf dataset has shown promising results and demonstrated the generalization capability of our proposed models for use in detecting other leaf-based diseases.

A potential future work includes deploying our light-weighted models in embedded devices. Our models, namely Our Model-112 and Our Model-128 require around 2.27 and 2.99 million parameters, and 7.6MB and 11.7MB of storage space to store the trained parameters, respectively. Both of these properties of our models enable them to realise on the resource constrained devices, such as mobile phones and tablets. Further, network parameter quantization [34], [35], and pruning [34] are some of the techniques that can be used to further compress the deep models. We will explore how these techniques can be incorporated with our framework to further compress the models, and formulate an application to deploy on rather limited resource constrained devices, such as low end smart phones.

## REFERENCES

- [1] UNCTAD. (2004). *Commodity Atlas Citrus Fruits*. Accessed: May 19, 2020. [Online]. Available: [https://unctad.org/en/Docs/ditcom20041ch3\\_en.pdf](https://unctad.org/en/Docs/ditcom20041ch3_en.pdf)
- [2] A. Urbaneja, T. G. Grout, S. Gravena, F. Wu, Y. Cen, and P. A. Stansly, "Citrus pests in a global world," in *The Genus Citrus*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 333–348.
- [3] M. Zhang, S. Liu, F. Yang, and J. Liu, "Classification of canker on small datasets using improved deep convolutional generative adversarial networks," *IEEE Access*, vol. 7, pp. 49680–49690, 2019.
- [4] A. R. Mishra, D. Karimi, R. Ehsani, and W. S. Lee, "Identification of citrus greening (HLB) using a VIS-NIR spectroscopy technique," *Trans. ASABE*, vol. 55, no. 2, pp. 711–720, 2012.
- [5] S. Sankaran and R. Ehsani, "Detection of huanglongbing disease in citrus using fluorescence spectroscopy," *Trans. ASABE*, vol. 55, no. 1, pp. 313–320, 2012.
- [6] J. Abdulridha, O. Batuman, and Y. Ampatzidis, "UAV-based remote sensing technique to detect citrus canker disease utilizing hyperspectral imaging and machine learning," *Remote Sens.*, vol. 11, no. 11, p. 1373, Jun. 2019.
- [7] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Comput. Electron. Agricult.*, vol. 150, pp. 220–234, Jul. 2018.
- [8] W. Pan, J. Qin, X. Xiang, Y. Wu, Y. Tan, and L. Xiang, "A smart mobile diagnosis system for citrus diseases based on densely connected convolutional networks," *IEEE Access*, vol. 7, pp. 87534–87542, 2019.
- [9] Q. H. Cap, H. Uga, S. Kagiwada, and H. Iyatomi, "LeafGAN: An effective data augmentation method for practical plant disease diagnosis," 2020, *arXiv:2002.10100*. [Online]. Available: <http://arxiv.org/abs/2002.10100>
- [10] Q. Wu, Y. Chen, and J. Meng, "DCGAN-based data augmentation for tomato leaf disease identification," *IEEE Access*, vol. 8, pp. 98716–98728, 2020.
- [11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
- [12] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: <http://arxiv.org/abs/1704.04861>
- [13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
- [14] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.
- [15] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*. [Online]. Available: <http://arxiv.org/abs/1905.11946>
- [16] K. Kc, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Comput. Electron. Agricult.*, vol. 165, Oct. 2019, Art. no. 104948.
- [17] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, Lille, France, vol. 2, 2015, pp. 1–8.
- [18] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087.
- [19] O. Rippel, M. Paluri, P. Dollár, and L. Bourdev, "Metric learning with adaptive density discrimination," 2015, *arXiv:1511.05939*. [Online]. Available: <http://arxiv.org/abs/1511.05939>
- [20] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Brief*, vol. 26, Oct. 2019, Art. no. 104340.
- [21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.
- [22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [23] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016.
- [24] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," *Symmetry*, vol. 10, no. 1, p. 11, Dec. 2017.
- [25] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers Plant Sci.*, vol. 8, p. 1852, Oct. 2017.
- [26] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3630–3638.
- [27] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'siamese' time delay neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 737–744.
- [28] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.
- [29] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1735–1742.
- [30] W. Fang, Y. Ding, and F. Zhang, "A novel convolution neural network for background segmentation recognition," in *Proc. Int. Conf. Cloud Comput. Secur. Cham, Switzerland: Springer*, 2018, pp. 172–182.
- [31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>

- [32] G. Hu. (2019). *Data for: A Low Shot Learning Method for Tea Leaf's Disease Identification*. [Online]. Available: <https://data.mendeley.com/datasets/dbjyfkn6jr/1>
- [33] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
- [34] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*. [Online]. Available: <http://arxiv.org/abs/1510.00149>
- [35] P. Gysel, M. Motamedi, and S. Ghiasi, "Hardware-oriented approximation of convolutional neural networks," 2016, *arXiv:1604.03168*. [Online]. Available: <http://arxiv.org/abs/1604.03168>



**SIVASUBRAMANIAM JANARTHAN** received the B.Sc. degree in engineering from the University of Moratuwa, Sri Lanka, in 2009, and the M.Tech. degree in software engineering from the National University of Singapore, in 2015. He is currently pursuing the Ph.D. degree with the School of Information Technology, Deakin University, Geelong, Australia. He worked as an IT Professional in various roles for a decade, till 2019. His current research interests include precision agriculture, computer vision, 3-D reconstruction, and machine learning.



**SELVARAJAH THUSEETHAN** received the B.Sc. degree from the University of Jaffna, Sri Lanka, in 2013. He is currently pursuing the Ph.D. degree with the School of Information Technology, Deakin University, Geelong, Australia. He has been a Probationary Lecturer with the Department of Computing and Information Systems, Sabaragamuwa University of Sri Lanka, since 2015. His current research interests include emotion recognition, computer vision, machine learning, and pattern recognition.



**SUTHARSHAN RAJASEGARAR** (Member, IEEE) received the Ph.D. degree from The University of Melbourne, Melbourne, VIC, Australia, in 2009. He was a Research Fellow with the Department of Electrical and Electronic Engineering, The University of Melbourne, and a Researcher in machine learning with the National ICT Australia. He is currently a Senior Lecturer with the School of Information Technology, Deakin University, Geelong, Australia. His current research interests include anomaly/outlier detection, distributed machine learning, pattern recognition, computer vision, precision agriculture, health analytics, and the Internet of Things (IoT).



**QIANG LYU** received the Ph.D. degree in food sciences from Jiangsu University, China, in 2010. He is currently an Associate Professor with the Citrus Research Institute, Southwest University, China. His research interests include image processing, spectroscopy, and remote sensing in agriculture.



**YONGQIANG ZHENG** received the B.Sc. and M.Sc. degrees in horticulture from Shandong Agricultural University, China, in 2001 and 2004, respectively, and the Ph.D. degree in botany from the Chinese Academy of Sciences, Beijing, China, in 2008. He is currently a Professor with the Citrus Research Institute, Southwest University, China. He is also the Deputy Director of the Key Laboratory of Citrus, Chinese Academy of Agricultural Sciences, the Leader of the Citrus Cultivation Physiology and Intelligent Agriculture Research Group, and the Scientist of the Intelligent Production of the National Modern Agriculture (Citrus) Industrial Technology System. His research interests include citrus cultivation physiology and intelligent agricultural technology.



**JOHN YEARWOOD** (Member, IEEE) received the B.Sc. degree from Monash University, Australia, the M.Sc. degree from The University of Sydney, Australia, and the Ph.D. degree from RMIT University, Australia. In 1989, he was a Lecturer with the School of Information Technology and Mathematical Science, University of Ballarat, Australia, where he was appointed as a Professor, in 2007. He is currently a Professor and the Head of the School of Information Technology, Deakin University, Australia. He has authored over 140 refereed journals, book chapters, and conference papers. His research interests include modern optimization theory and techniques and their applications in pattern recognition, signal processing, and decision support systems.

...