Computers and Electronics in Agriculture 211 (2023) 107967

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

CLA: A self-supervised **c**ontrastive learning method for **l**eaf disease identification with domain **a**daptation

Ruzhun Zhao [a], Yuchang Zhu [b], Yuanhong Li [c,*]

[a] School of Automobile, Guangdong Mechanical & Electrical Polytechnic, Guangzhou, Guangdong 510515, China
[b] College of Engineering, South China Agricultural University, Guangzhou, Guangdong 510642, China
[c] College of Electronic Engineering, South China Agricultural University, Guangzhou, Guangdong 510642, China

ARTICLE INFO

Keywords:
Deep learning
Leaf disease identification
Self-supervised learning
Domain adaptation

ABSTRACT

Plant leaf diseases cause a decrease in crop yield and degrade the quality, which presents the urgent need for leaf disease identification. Recently, deep learning technologies, especially computer vision, have emerged as a powerful tool in plant leaf disease identification. However, existing methods invariably rely on large-scale labeled data for model training. Although self-supervised learning which uses large-scale unlabeled data for pre-training provides a useful scheme, these unlabeled data are messy, e.g., images with different shooting angles and backgrounds. In this regard, such data results in distribution-shifted training datasets, which degrades model performance. To address these problems, we propose a self-supervised **C**ontrastive learning method for **L**eaf disease identification with domain **A**daptation (CLA), including pre-training with forwarding and fine-tuning with domain adaptation stage. Specifically, CLA utilises large-scale yet messy unlabeled data to train the encoder and obtains their visual representations in the pre-training stage. Based on small labeled data, CLA trains the domain adaptation layer (DAL) and the classifier in the fine-tuning stage. Due to the DAL, CLA can align labeled data and unlabeled data in the fine-tuning stage and generates more general visual representations, which improves the domain adaptation ability of CLA. Experiments are conducted to evaluate the performance of CLA and demonstrate that CLA outperforms the comparison methods by a large margin on both domain adaptation and accuracy performance, with the highest accuracy of 90.52%. To better understand our proposed method, additional experiments are also conducted to explore the influencing factors of CLA.

1. Introduction

1.1. Motivation

Plant diseases seriously affect crops' growth and fruiting, resulting in the decline of crop yields and quality, which cause economic losses (Jogekar & Tiwari, 2021; Shantkumari & Uma, 2021). According to the statistics (Gupta et al., 2019; Tang et al., 2020), 20%-40% of the decline in crop yields worldwide is caused by plant diseases. As shown in Fig. 1 (a), plant leaf diseases are diverse, including rot, rust, spot, etc (Annabel et al., 2019). Due to the impact of plant diseases, the appearance of plant leaves changes as shown in Fig. 1(b). By observing the appearance of leaves, many plant diseases can be diagnosed at an early stage, and then effective treatment can be carried out (Dwivedi et al., 2021; L. Li et al., 2021). Traditionally, diagnosing diseases by human experts is slow, laborious, and easily misdiagnosis (Ngugi et al., 2021; Verma et al., 2020) due to the diversity of plant diseases. Motivated by the great success of deep learning in various applications (Hu et al., 2022), researchers have applied deep learning models to the identification of plant leaf diseases, which reduces the cost of plant leaf disease identification and makes identification faster in monitoring large crop farms (Ashwinkumar et al., 2022). However, due to the complexity of plant leaf disease identification and the scarcity of samples, there are still many problems remaining under-explored.

Advances in plant leaf disease identification using deep learning technologies can be divided into three categories, i.e., identification model improvement (IMI) (Bansal et al., 2021; Lu et al., 2022), few-shot learning (FSL) (J. Yang et al., 2022), and self-supervised learning (SSL) (Güldenring and Nalpantidis, 2021). In addition, there are some studies that we do not discuss because these studies pay more attention to augmenting plant leaf disease datasets through GAN (Zhang et al., 2022a; Zhao et al., 2021; Zhou et al., 2021). Specifically, IMI aims to

* Corresponding author at: College of Electronic Engineering, South China Agricultural University, Guangzhou, Guangdong 510642, China.
E-mail addresses: zhaoruzhun@gdmec.edu.cn (R. Zhao), zhu_yuchang@stu.scau.edu.cn (Y. Zhu), liyuanhong@stu.scau.edu.cn (Y. Li).

https://doi.org/10.1016/j.compag.2023.107967
Received 31 August 2022; Received in revised form 21 April 2023; Accepted 2 June 2023
Available online 9 June 2023
0168-1699/© 2023 Elsevier B.V. All rights reserved.

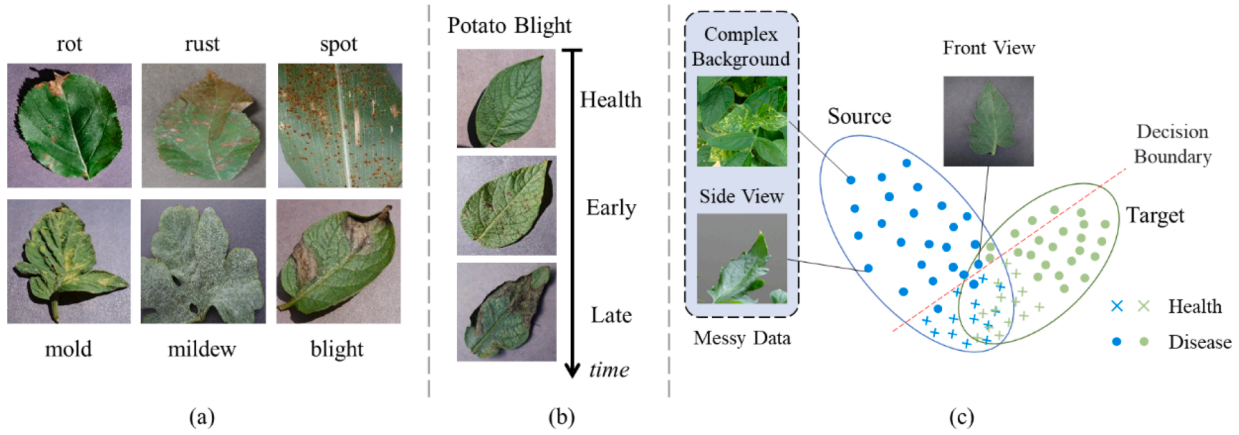

Fig. 1. An overview of plant leaf diseases and the scenario in this work. (a) some plant leaf disease samples. (b) plant leaf disease stages. (b) an overview of the scenarios that our approach focuses on.

develop an identification model for plant leaf disease that exhibits superior performance, leveraging models based on convolutional neural networks (CNNs) (LeCun et al., 1998) or transformers (Wang et al., 2021). Despite the high accuracy achieved by these methods in plant disease identification, they often rely on supervised training. To address the issue of limited labeled data, FSL trains a model to recognize similarities between samples and transfers this knowledge to the target domain with a small number of labeled samples known as the support set. However, FSL's performance significantly degrades when the source domain dataset is dissimilar to the support set. SSL is a training approach that involves pre-training a model using a large volume of unlabeled data, followed by fine-tuning it with a small amount of labeled data based on downstream tasks. This approach enables efficient utilization of unlabeled data, but it is noteworthy that only a few studies have explored the application of self-supervised learning in leaf disease identification.

In summary, the current research on leaf disease identification mainly focuses on identification model improvement and few-shot learning, both of which typically require a certain amount of labeled data for training, which contradicts the scarcity of labeled leaf disease datasets. As described in Güldenring and Nalpantidis (2021), it is feasible to collect a large number of unlabeled images specific to agricultural scenarios. Inspired by SSL methods (Chen et al., 2020b; Grill et al., 2020; He et al., 2020), these unlabeled data can be used as pre-training data and then the small labeled data is required to fine-tune networks. However, as shown in Fig. 1(c), we observe that numerous unlabeled data (source domain) collected in real-world scenarios may be messy due to different shooting views or complex backgrounds, e.g., complex background images and side shooting view images. This leads to a distribution shift between numerous unlabeled data and small labeled data (target domain) for fine-tuning, hurting model performance (Cole et al., 2022). To prevent model performance from the distribution shift, it is necessary to make the model domain adaptive.

1.2. Related work

In IMI, researchers have been primarily focusing on enhancing the accuracy of plant disease identification models using Convolutional Neural Networks (CNNs) or Transformers (Rayhan & Setyohadi, 2021; R. Yang et al., 2021). For CNNs-based models, Ji et al. (2020) proposed a unified framework to enhance feature extraction by combining multiple CNNs, thereby improving the accuracy of leaf disease identification to 98.57% on Plantvillage dataset; Zhang et al. (2021) improved the accuracy of leaf disease identification by using a multi-activation function module and a preprocessing method to enhance disease samples, achieving an accuracy of 97.41% on their own dataset. However, it should be noted that the superior performance of these models is largely due to the high similarity between the training and test datasets (J. Lu et al., 2021; Mohanty et al., 2016), which may be attributed to improper dataset division. For Transformer-based models, Hee-Jin et al. (2020) propose a new deep learning architecture for leaf disease identification that considers the leaf spot attention mechanism. F. Wang et al. (2022a) develope a backbone network based on Swin Transformer to improve the augmentation and recognition of practical cucumber leaf diseases. While these approaches show promise, it is important to note that detection Transformers can suffer from performance drops when trained on small-sized datasets (Wang et al., 2022b). It is worth noting that some studies have focused on both improving the accuracy of leaf disease identification models and making them easy to deploy on computationally limited platforms. For instance, Falaschetti et al. (2021) implemented a low-cost, low-power, and real-time image detector based on compressed CNNs. Zinonos et al. (2021) combined long-range technology and deep learning technology to overcome high computational costs. Too et al. (2019) focused on fine-tuning and found a simpler and more effective model framework through comparative studies. However, previous studies still rely on large-scale labeled leaf disease data and may perform poorly in real-world scenarios. Even with transfer learning techniques, such as those explored by (Chen et al., 2020a), datasets containing thousands of annotated leaf disease images are still required in the fine-tuning stage.

In FSL, researchers focus on how learners can transfer information to related domains, such as leaf disease, by conducting inferences on unseen data with the help of a support set containing a limited quantity of labeled data (Argüeso et al., 2020; Y. Wang & Wang, 2021). For instance, Jadon (2020) proposed a metrics-based few-shot learning framework for low-resolution leaf disease identification using a matching network to map data from the source domain (training set) and the target domain (test set) to the same embedding space to increase transferability. To develop a model with transferability, L. Chen et al. (2021) also employed the concept of matching. Yan et al. (2021) mixed source domain and target domain images and aligned the mixed images with the target domain images using a subdomain alignment mechanism to transfer the knowledge learned by the model. Although the above-mentioned FSL techniques primarily concentrate on the transferability of models, the datasets they used demonstrate a strong correlation between the training set and the test set. Therefore, when there is no association between the training set and the test set, the accuracy of the model will drop dramatically. Furthermore, most existing FSL methods for leaf disease identification employ a supervised learning scheme for model training, which may cause the model to miss the opportunity to learn from more relevant unlabeled data. To address this issue, Y. Li & Chao (2021) proposed a few-shot learning method based on semi-

supervised learning. This method selects unlabeled leaf disease samples through the confidence interval and assigns pseudo labels to these samples, allowing the model to learn from unlabeled data. Additionally, to gain better insights into few-shot learning for leaf disease identification, some studies (Y. Li & Yang, 2021; Nuthalapati & Tunga, 2021) have focused on the baseline and dataset of leaf disease identification. In summary, FSL methods still require a certain amount of labeled leaf disease data to achieve good results in the field of leaf disease identification. Meanwhile, the transferability of the model still depends on a high correlation between the training set and test set, such as both being images of plants.

SSL is a deep learning scheme that has emerged in recent years, with classic models such as MoCo (He et al., 2020), SimCLR (Chen et al., 2020b), BYOL (Grill et al., 2020), SwAV (Caron et al., 2020), and SimSiam (X. Chen & He, 2021). With the efforts of researchers, SSL methods have outperformed supervised learning methods and have been applied to various tasks (Breiki et al., 2021) and fields, including agriculture (Najafian et al., 2021), biology (Bommanapally et al., 2021), aviation (Wen et al., 2021). While the application of SSL in agriculture is still in its infancy, it holds great promise due to the scarcity of labeled datasets in agriculture (Güldenring and Nalpantidis, 2021). In the agricultural field, a large number of unlabeled agricultural images are available and can be utilized in the pre-training stage of SSL. For instance, Kar et al. (2021) used more than 9000 unlabeled images for pre-training and a small number of labeled images for fine-tuning to classify 12 types of agricultural pests using the BYOL method with 94% accuracy on a custom dataset collected from the research fields of Iowa State University, USA. Güldenring et al. (2021b) explored the applicability of self-supervised contrastive learning on agricultural images. Fang et al. (2021) used the cross-iterative under-clustering algorithm based on kernel k-means to provide pseudo labels for unlabeled data to achieve self-supervised learning and plant disease classification, without using the current mainstream self-supervised learning framework. Although the SSL method in plant leaf disease identification remains unexplored, it holds significant potential in this field. By leveraging large amounts of unlabeled data, SSL can address the shortage of labeled data and improve the accuracy of leaf disease identification models.

1.3. Contributions

To address the above problems, this paper is devoted to leaf disease identification with few labeled training data and messy unlabeled training data. Specifically, we propose a self-supervised contrastive learning method with domain adaptation, namely CLA, and conduct experimental verification. The contributions of this article are as follows:

(1) We study a novel problem for leaf disease identification on the scenarios of limited labeled data and distribution shift. To the best of our knowledge, this is the first work to utilize the contrastive learning with domain adaptation to explore leaf disease identification in this particular scenario.
(2) To address these problems, we propose a self-supervised **C**ontrastive learning method for **L**eaf disease identification with domain **A**daptation (CLA). Specifically, CLA involves two phases: pre-training and fine-tuning. During the pre-training phase, a substantial amount of unlabeled data is used to pre-train the encoder, which helps to address the issue of limited labeled data. In the fine-tuning phase, the domain adaptation layer (DAL) is introduced to mitigate the problem of distribution shift, allowing the model to be fine-tuned on a small amount of labeled data from the downstream task.
(3) We conduct enormous experiments to evaluate of the proposed method and explore the impact of factors such as dataset volume and components. Experimental results demonstrate the effectiveness of CLA for leaf disease identification. Our experiments will provide useful inspiration and experience for future research.

The following sections are organized as follows: Section 2 describes the details of CLA and the dataset; Section 3 conducts experiment and analyzes experimental results; Section 4 summarizes the work of this article.

2. Methodology

2.1. Problem setting

In this work, we focus on leaf disease identification with small labeled data and large-scale unlabeled data as training data. We aim to learn a model F to predict which plant species and diseases there exist with the leaf image x as input, which can be regarded as a N classification task T. Here, N is the number of predefined categories and can be denoted by "plant species & disease/healthy", such as apple black rot, apple health, tomato target spot, etc. Specifically, given an unlabeled source domain dataset $D_s = \{x_s^i\}_{i=1}^n$ consists of n images, and a fully-labeled target domain dataset $D_t = \{(x_t^j, y_t^j)\}_{j=1}^m$ consists of m images and label pairs, both $\{x_s^i\}$ and $\{x_t^j\}$ belong to the same set of N predefined categories. Here, x_s^i represents the i-th sample in the source domain dataset, x_t^j represents the j-th sample in the target domain dataset and y_t^j is the label of the j-th sample. Assume that model F consists of the encoder f and the classifier c. According to our setting, f is pre-trained on D_s according to the contrastive learning paradigm and c is fine-tuned on D_t. Given the input x which is related to D_s or D_t, the trained model F aims to make a classification and then outputs results $\widehat{y}$. Thus, this problem can be summarized as: given D_s for pre-training, D_t for fine-tuning, for any input x, the trained model F makes high accuracy prediction, i.e., $F(x) \to \widehat{y}$.

However, there exists the distribution shift between D_s and D_t or between intra-class images of D_s. To further show the distribution shift, we introduce a dataset example in Fig. 2 with the dataset used in experiments of Section 4. Due to the scarcity of labeled plant leaf images, we use images with the front view in Plantvillage dataset (Hughes et al., 2015) as the target domain data, but in fact, target domain data can be leaf images with any shooting views or complex background. In this example, D_t comprises images of plant leaves captured in a controlled laboratory environment, depicting the leaves from the front and set against a simple background, as shown in the first row of Fig. 2. In contrast, D_s comprises images of plant leaves captured in various real-world scenarios, featuring different shooting views and complex backgrounds, as shown in the second and third rows of Fig. 2. Assume that x_s^i in D_s and x_t^j in D_t belong to the same class, but due to inconsistent shooting views, different backgrounds, etc., the trained model may misidentifies these two images as different categories. This is the distribution shift between these two datasets, resulting in poor performance of the trained model. In this work, we also aim to alleviate such distribution shift.

Note that the source domain and target domain in this work is different from that in domain adaptation. We use these two technical terms (i.e., the source domain and target domain) only for the convenience of distinguishing. In domain adaptation common protocol, the source domain dataset is labeled data and the target domain dataset is unlabeled data. The target domain represents the scenario in which a trained model will be deployed. In our work, label information in the source and target domain is opposite to the domain adaptation. Moreover, we do not limit the distribution of the target domain, which means that the distribution of the target domain may also be the same as the source domain.

Fig. 2. Samples of the source domain dataset and the target domain dataset. The first row are samples of the target domain. The second and third rows are samples of the source domain. Each column belongs to the same leaf disease.

2.2. CLA: The proposed framework

In this subsection, we introduce our proposed framework named CLA. First, we give an overview of CLA. Then, we present a detailed description of each component of CLA. Finally, we introduce the network settings of CLA used in experiments.

2.2.1. Overview

As shown in Fig. 3, CLA consists of two stages, i.e., pre-training with forwarding and fine-tuning with domain adaptation. In the pre-training with forwarding stage, CLA pre-trains the encoder with a large amount of unlabeled data (source domain) based on the contrastive learning paradigm, i.e., SimSiam (X. Chen & He, 2021). As shown in the left part of Fig. 3, there is an asymmetric structure with two sides to process two augmented images. One side is "an encoder plus stop gradient operation" and the other side is "an encoder plus a predictor". Here, the encoder on two sides is the same and will be used to extract features for the classifier after pre-training. The predictor is designed for the asymmetric structure and transforms the output vector of the encoder to match the output of the other side. Stop gradient operation means that the output vector of the encoder is treated as a constant during backpropagation and its gradient is not calculated. With the pre-trained encoder, forward propagation is performed to obtain the source embedding which optimizes the parameters of DAL in the fine-tuning stage. Here, DAL is employed to align the source domain with the target domain and improve the transferability of CLA.

In the fine-tuning with domain adaptation stage, CLA freezes the parameter update of the encoder and uses a small amount of labeled data (target domain) to train the DAL and classifier. Note that we do not limit the distribution of target domain data, which means the leaf images in the target domain may be any shooting view or complex background. With images in the target domain dataset as input, the trained encoder outputs the target embedding. Then, DAL processes the target embedding and outputs results with the same dimensions as the target embedding. For example, in our experiment setting, the dimension of target embedding is 2048 and the dimension of the output vector of DAL is also 2048. This process achieves the alignment of source embedding and target embedding to ensure that the visual representation for the classifier is valid. In CLA, the loss function mainly includes the domain alignment loss L_{MMD} and the downstream classification task loss L_c. The former ensures that the distribution of source embedding and target embedding is as similar as possible to provide an effective visual rep-

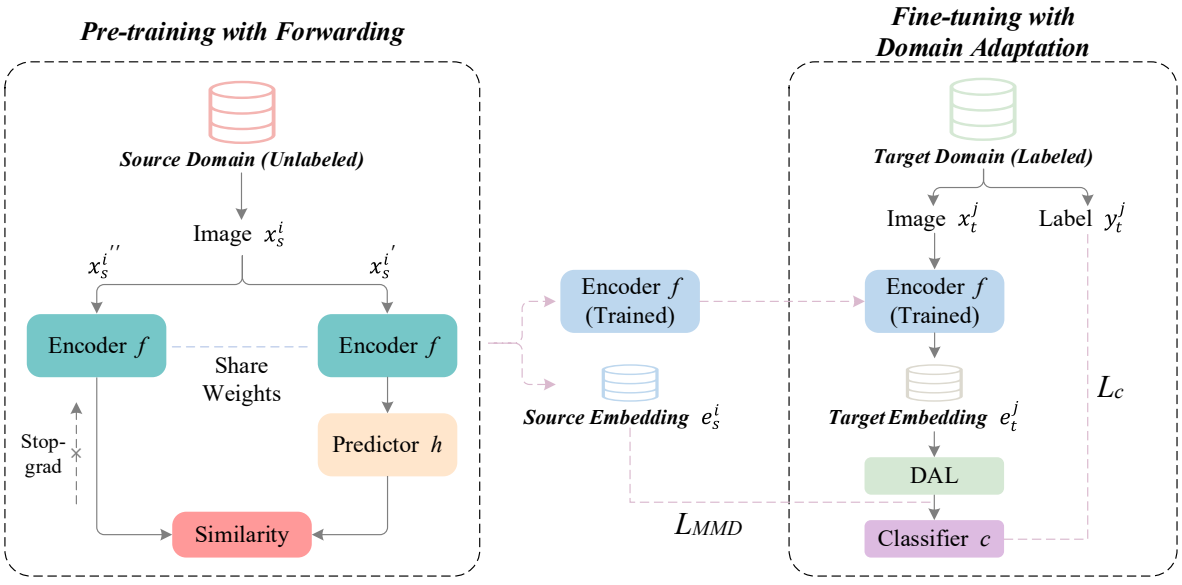

Fig. 3. Overview of CLA. (left) Pre-training with forwarding stage. Encoders f in two views are the same. (right) Fine-tuning with domain adaptation. Here, encoders f is the same as that in the pre-training stage.

resentation for the classifier, and the latter ensures the classification accuracy of the classifier.

2.2.2. Pre-training with forwarding

In the pre-training stage, CLA trains the encoder by contrastive learning and performs forward propagation after pre-training to obtain the source embedding, i.e., the vector representation of the source domain dataset. The purpose of contrastive learning is to train a model by reducing the representation distance between positive sample pairs and increasing the representation distance between negative sample pairs. According to the common protocol, contrastive learning takes an image and its augmented images as positive sample pair, and two different images in the dataset as negative sample pair. Specifically, for an image x_s^i in D_s, the rest of the images in D_s are denoted by x_s^k, where $k \neq i$. Both $x_s^{i'}$ and $x_s^{i''}$ are the augmented images of x_s^i. Here, (x_s^i, x_s^i) and $(x_s^i, x_s^{i''})$ are positive sample pairs, and (x_s^i, x_s^k) can be considered a negative sample pair. As shown in Fig. 1(c), the source domain data used in pre-training is relatively messy, which results in large differences between sample pairs. If the model is trained with the above idea, the contrastive task will be too simple, which will affect the performance of the trained encoder (Cole et al., 2022). To avoid this effect, CLA follows the SimSiam (X. Chen & He, 2021) contrastive learning framework which trains the encoder without negative sample pairs.

As shown in Fig. 3, the encoders of the two views are the same and share weights. the predictor h is used to make the two views achieve an asymmetric structure, i.e., different structures for two augmented images pass-through, thereby preventing the collapse of the model during training. Here, "model collapse" means that the trained model maps all inputted data to the same representation. Note that the predictor follows the setting in SimSiam (X. Chen & He, 2021), which is the identity mapping and is only used in the pre-training stage. In the pre-training stage: for any image x_s^i in D_s, we first perform random augmentation to obtain two augmented images of x_s^i, i.e. $x_s^{i'}$ and $x_s^{i''}$. After a forward with the asymmetric structure in the two views, we can obtain two output vectors, $p_1^i \triangleq h(f(x_s^{i'}))$ and $z_2^i \triangleq f(x_s^{i''})$. Likewise, by exchanging the two views, we can obtain $p_2^i \triangleq h(f(x_s^{i''}))$ and $z_1^i \triangleq f(x_s^{i'})$. Next, the goal of pre-training is to minimize the negative cosine similarity between the output vectors of the two views. With the output vectors of two views, the symmetrical loss is constructed according to BYOL (Grill et al., 2020), as shown in Equation (1):

$$L = \frac{1}{2} S(p_1^i, z_2^i) + \frac{1}{2} S(p_2^i, z_1^i) \tag{1}$$

Where $S \in [-1, 1]$ is the negative cosine similarity, measuring the similarity between two vectors. The higher the similarity, the smaller the value of the negative cosine similarity. The calculation formula is shown in Equation (2):

$$S(p_1^i, z_2^i) = - \frac{p_1^i}{\|p_1^i\|_2} \cdot \frac{z_2^i}{\|z_2^i\|_2} \tag{2}$$

To further prevent model collapse during training, SimSiam performs the stop gradient operation to stop backpropagation for the side without the predictor. With the stop gradient operation, the vector output by this side will be treated as the constant and its gradient is not calculated. CLA follows the idea of stop gradient in SimSiam. Thus, denoting the stop gradient operation as *stop-grad*, the loss function can be summarized as:

$$L = \frac{1}{2} S(p_1^i, stopgrad(z_2^i)) + \frac{1}{2} S(p_2^i, stopgrad(z_1^i)) \tag{3}$$

After completing the pre-training, we obtain the optimized parameters θ of the encoder. Then, we need to forward the dataset once with θ to obtain a low-dimensional vector representation of the source domain dataset for domain distribution alignment, as shown in Equation (4).

$$e_s^i = f_\theta(x_s^i) \tag{4}$$

Where f_θ is the encoder with the optimal parameters; e_s^i is the low-dimensional feature representation of the source domain data.

2.2.3. Fine-tuning with domain adaptation

In the fine-tuning stage, CLA freezes the parameters of the encoder and uses a small number of labeled target domain datasets to fine-tune. Due to the limited computing power in agriculture, we only update the parameters of DAL and the classifier in fine-tuning stage, which is different from the common protocol of self-supervised learning (Chen et al., 2020b). Here, DAL is sued to alleviate the distribution shift between the source domain dataset D_s and the target domain dataset D_t in the fine-tuning stage, so that the classifier performs better. Inspired by domain adaptation (Borgwardt et al., 2006; Tzeng et al., 2014), we construct DAL as a linear network, such as a fully connected layer, and add it between the encoder and the classifier. Thus, DAL transforms target embedding e_t^j, i.e., the output of the trained encoder for the target domain, to match source embedding e_s^i and learns a representation that can be effectively discriminated by the classifier. To train the optimal parameters of DAL, maximum mean discrepancy (MMD) is employed as one of the loss terms to measure the representation distance between the output vector of DAL and e_s^i. The goal of MMD learns a DAL which generates the representation of the target domain that is as close as possible to the source domain.

Assume that the pre-trained encoder with optimal parameters is f_θ, DAL with the same input and output size is g, the output size of the classifier is the number of categories that need to be classified. With the fixed f_θ, CLA trains the parameters of g and c on D_t, which can be summarized as: for any image x_t^j and its label y_t^j in D_t, after being processed by the encoder, we obtain feature vectors $z_t^j = f_\theta(x_t^j)$, and then transform these feature vectors through DAL to obtain aligned feature vectors $\widehat{z}_t^j = g(z_t^j)$. Finally, the aligned feature vector $\widehat{z}_t^j$ is input to the classifier to get a prediction $\widehat{y}_t^j = c(\widehat{z}_t^j)$ of plant type and its disease. To guide the parameters updating of g and c, we construct the multi-task loss, which includes the distribution distance loss L_{MMD} and the classification loss L_c, and its formula is as follows:

$$L_{total} = L_c + \alpha L_{MMD} \tag{5}$$

In Equation (5), L_c is the cross-entropy loss function, as shown in Equation (6); $\alpha \in [0, 1]$ is the coefficient of the distribution distance loss, representing the important weight of L_{MMD} in the multi-task loss; L_{MMD} measures the distance between the two distributions (i.e., the source embedding e_s^i and the aligned feature vector $\widehat{z}_t^j$ of the target domain) by MMD. Specifically, MMD is to map two distributions into the same vector space to calculate the distance of two distributions. This process is to calculate the difference between the expectations of the two distributions after mapping, which is called mean discrepancy. The suprema of mean discrepancy are the maximum mean discrepancy. If the mapping function is represented by $\phi(\cdot)$, it is shown in Equation (7).

$$L_c = - \sum_{j=1}^{m} y_t^j \log \widehat{y}_t^j \tag{6}$$

$$L_{MMD} = \left\| \frac{1}{n} \sum_{i=1}^{n} \phi(e_s^i) - \frac{1}{m} \sum_{j=1}^{m} \phi(\widehat{z}_t^j) \right\| \tag{7}$$

2.2.4. Network settings

In this subsection, we provide the network settings of CLA used in our experiments, including the structure of the encoder, predictor, DAL, and classifier, as shown in Table 1. Note that this network setting does not represent the optimal network structure, but only the baseline settings.

In Table 1, BN is the abbreviation of batch normalization, FC is the

Table 1
Network architecture. The inference network consists of the encoder, DAL and the classifier. The predictor is only used in the pre-training stage.

Network component	Architecture	Input	Output
Encoder	An encoder consists of a backbone (ResNet 18 (He et al., 2016)) and a projection head (The MLP includes 3 FC layers with the BN operation)	ResNet 18: 224 × 224 × 3 Projection head: 2048	ResNet 18: 2048 Projection head: 2048
Predictor	The MLP includes 2 FC layers. All layers except the output FC layer have the BN operation	2048	2048
Domain adaptation layer (DAL)	1 FC layer	2048	2048
Classifier	1 FC layer with bias	2048	38

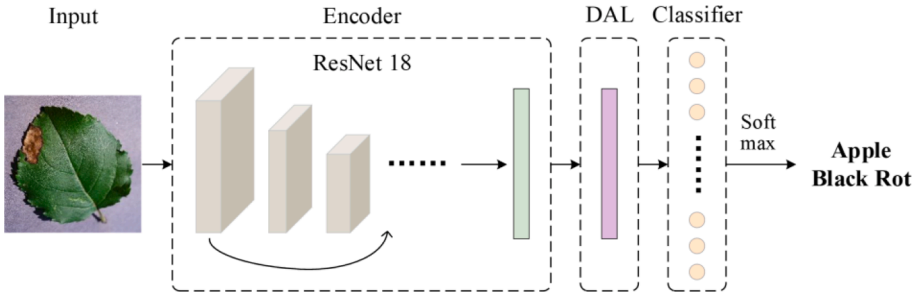

Fig. 4. An inference case with our network settings.

Table 2
Statistics of the experimental datasets.

Dataset name	Class number	Image number
Plantvillage	38	54,303
PlantDoc	27	2570
Augmented leaf disease dataset	32	4827

abbreviation of fully connected, and the maximum value of the output of the classifier is the classification result. ResNet (He et al., 2016) is a well-known deep learning network with excellent performance and small parameters proposed by He Kaiming. The residual learning unit is the most important component of ResNet. By stacking residual units, the neural network model can become deeper, and solve the degradation problem, i.e., the network accuracy saturates or even declines when the network depth increases. Specifically, due to the limited computation, the backbone of CLA is ResNet 18 with 18 layers. Note that ResNet 18 is not the only choice of backbone. Other architectures, such as ResNet50, maybe perform better, but it requires a higher computation cost.

After CLA is trained, we combine the encoder, DAL, and classifier in order. As shown in Fig. 4, the trained Encoder, DAL, and classifier are combined to perform inference to detect the disease in plant leaves.

3. Datasets

In the experiment, we used 3 datasets, namely Plantvillage[1] (Hughes et al., 2015), PlantDoc[2] (Singh et al., 2020), and Augmented leaf disease dataset[3] (Hewarathna, 2022). In Plantvillage, 3800 images were used as the target domain dataset for the fine-tuning stage, 3800 images were used as the test set, and the rest were used as the source domain dataset for the pre-training phase. Due to the scarcity of labeled data, we have to utilize the part of data in Plantvillage dataset as the target domain and the test set. PlantDoc and Augmented leaf disease dataset datasets are plant leaf images in real-world scenarios. We combine these two datasets to simulate the distribution shift data with different shooting views and complex backgrounds. The combined dataset is called a synthetic dataset. The synthetic dataset is mainly added to the source domain dataset or the target domain dataset as distribution shift data. In the experiment, we added the synthetic dataset into the Plantvillage and formed source domain or target domain datasets with different mix ratios. Specifically, denoting the number of images of Plantvillage as n_{pv} and denoting the number of images of the synthetic dataset as n_{sd}, we calculate the mix ratio of the dataset according to Equation (8).

$$MR = \frac{n_{sd}}{n_{sd} + n_{pv}} \times 100\% \tag{8}$$

In the experiments, we only use the 38 categories defined by Mohanty et al. (2016) and remove the data which is irrelevant to the 38 categories from the dataset. Table 2 shows the statistics of each dataset after removing it. More detailed information is described in Table 9 of Appendix B.

3.1. Plantvillage dataset

Plantvillage (Hughes et al., 2015) is currently the most popular public dataset in the field of plant leaf diseases, with 54,303 256 × 256 RGB images of plant leaves (some works published that Plantvillage has 54,306 images, and the dataset we obtained from here[4] is 54,303 images). Plantvillage contains 14 crops and 26 diseases, which are annotated into 38 categories (Mohanty et al., 2016), see Table 9 for details. As shown in Fig. 5, we divided the Plantvillage dataset into 3 parts, each containing all 38 categories. Specifically, Part 1 contains 47,003 images to simulate the unlabeled source domain dataset. Due to the relatively small number of images of Apple Cedar apple rust and Potato healthy categories, we extracted 100 images and 200 images from the corresponding categories in the New Plant Diseases Dataset[5] to join these two categories. Part 2 contains 3800 images, 100 images per category, which is regarded as the labeled target domain dataset. Part 3 contains 3800 images, 100 images per category, which is regarded as the test set to evaluate model performance.

3.2. Synthetic dataset

We constructed a synthetic dataset by combining the PlantDoc

[1] https://github.com/spMohanty/PlantVillage-Dataset.
[2] https://github.com/pratikkayal/PlantDoc-Dataset.
[3] https://www.kaggle.com/datasets/asheniranga/augmented-leaf-dataset.
[4] https://github.com/spMohanty/PlantVillage-Dataset.
[5] https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset.

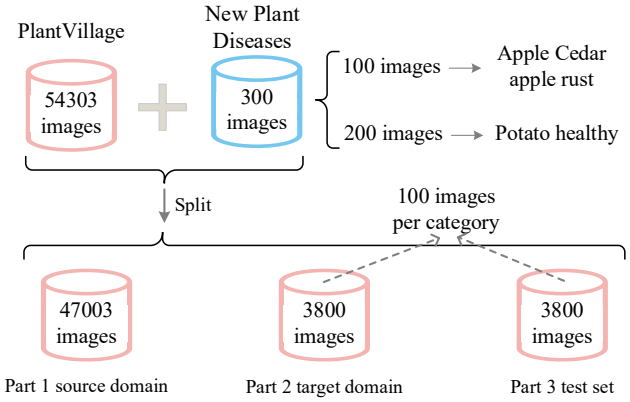

Fig. 5. The compositions and splits of Plantvillage dataset.

dataset (Singh et al., 2020) and the Augmented leaf disease dataset (Hewarathna, 2022), but the synthetic dataset only included 37 categories (Peach Bacterial spot missing). The synthetic dataset is mainly used to simulate the distribution shift data of the source domain dataset, i.e., plant leaf images with different shooting angles or complex backgrounds. There are a total of 7397 RGB images in the synthetic dataset, of which 2570 are from PlantDoc and the rest are from the Augmented leaf disease dataset. The following is a detailed introduction to the PlantDoc dataset and Augmented leaf disease dataset:

PlantDoc (Singh et al., 2020) is a non-lab dataset of plant leaf disease collected from real-world scenarios, with 2598 RGB images (2570 in our statistics), including 13 crops and 27 categories. Since many public datasets are in a controlled environment (uniform lighting, simple background, frontal shooting), the performance of the trained model dramatically decreases when applied to real-world scenarios. PlantDoc provides training data that is closer to real-world scenarios to obtain a better leaf disease identification model.

The augmented leaf disease dataset (Hewarathna, 2022) added the virtual background to plant leaf images in an augmented method, so that the dataset is closer to real-world scenarios. The augmented leaf disease dataset includes 8567 256 × 256 RGB images covering 38 categories, but only 32 categories belong to the 38 categories we used. Therefore, we only used 4827 plant leaf images in this dataset and removed images of 6 categories that were not relevant to the experiment. We show the statistics data of the augmented leaf disease dataset after removing it in Table 2.

4. Experimental results and discussion

We conducted experiments to evaluate the performance of CLA. The purpose of the experiments can be summarized in three aspects: 1) To verify the effectiveness of CLA in leaf disease identification and its superiority compared to other methods, as shown in Section 4.2. 2) To validate the effectiveness of DAL for domain shift scenarios, as shown in Section 4.3. 3) To study the impact of influencing factors on the performance of CLA, as shown in Section 4.4. Moreover, we present more detailed experimental results about influencing factors in Appendix A.

4.1. Experimental settings

4.1.1. Metrics

We use Precision (*P*), Recall (*R*), F1-score, and Accuracy as evaluation metrics. Due to the multi-classification task, these four metrics are calculated and then averaged by the macro method, i.e., calculating metrics for each category and then conducting arithmetic mean for each metric. Note that, in the multi-classification task, the *P*, *R*, and F1-scores calculated by the micro method have the same value and equal to the accuracy. Here, the micro method is calculating metrics globally by counting the total true positives, false negatives, and false positives. Thus, it can be considered that we calculated the *P*, *R*, and F1-score using both macro and micro methods. The calculation details of these four metrics are as follows.

For *P* and *R*, the computing method of macro averaged first calculates the metrics of each category separately and then develops an unweighted average of metrics to obtain macro averaged results. For the F1-score, Macro-Precision (Macro-P) and Macro-Recall (Macro-R) are calculated first, and then Macro-F1 is obtained according to Equation (13). Assuming that *TP* represents true positive, *FP* represents false positive, *TN* represents true negative, *FN* represents false negative. Specifically, for the *N* classification task, we regard each class as a positive class and the rest as a negative class, so we can count *N* numbers of $\{TP_k\}_{k=1}^{N}$, $\{FP_k\}_{k=1}^{N}$, $\{TN_k\}_{k=1}^{N}$, $\{FN_k\}_{k=1}^{N}$. For each category, we can calculate *P* and *R* as follows.

$$P_k = \frac{TP_k}{TP_k + FP_k} \tag{9}$$

$$R_k = \frac{TP_k}{TP_k + FN_k} \tag{10}$$

Next, we take a weighted average of P_k and R_k to get Macro-P and Macro-R, as shown below.

$$Macro - P = \frac{1}{k}\sum_{k=1}^{N} P_k \tag{11}$$

$$Macro - R = \frac{1}{k}\sum_{k=1}^{N} R_k \tag{12}$$

Based on Macro-P and Macro-R, we can calculate Macro-F1 as shown below.

$$Macro - F1 = \frac{2{*}Macro - P{*}Macro - R}{Macro - P + Macro - R} \tag{13}$$

Accuracy is to evaluate the prediction accuracy of the model, which is calculated as the ratio of the number of correctly predicted samples to the number of all samples in the prediction.

4.1.2. Comparison methods

Under the same backbone (ResNet18), we explore different training strategies for comparison, including supervised, ImageNet pre-training, and SimSiam contrastive learning. *Supervised Method (SM)*: Training model from scratch on the labeled dataset which is the same as the fine-tuning stage of the CLA framework, and optimizing the backbone and classifier weight parameters. *ImageNet Pre-trained Backbone-Linear (INPB-L)*: Based on the ResNet 18[6] wt parameters pre-trained on ImageNet, the parameters of the backbone are frozen, and the same dataset as the fine-tuning stage of the CLA framework is used to train a classifier with one fully connected layer. *SimSiam-Linear* (X. Chen & He, 2021) *(S-L)*: Based on the same dataset as the pre-training stage of CLA, the backbone is pre-trained using the SimSiam framework, and a single-layer fully connected layer classifier is trained through *INPB* strategy. The illustration of these three comparison methods is shown in Fig. 6.

4.1.3. Model training

All models were trained from scratch, except for models that used ImageNet pre-trained weights. Due to limited computation, the backbone used in all our experiments is ResNet18, while the ResNet50 backbone (Caron et al., 2020; Chen et al., 2020b; He et al., 2020) is commonly used in contrastive learning, which may secure a better accuracy. We optimize the model parameters by stochastic gradient descent (SGD) optimizer for all methods, the weight decay is 0.0005 and

[6] https://download.pytorch.org/models/resnet18-5c106cde.pth.

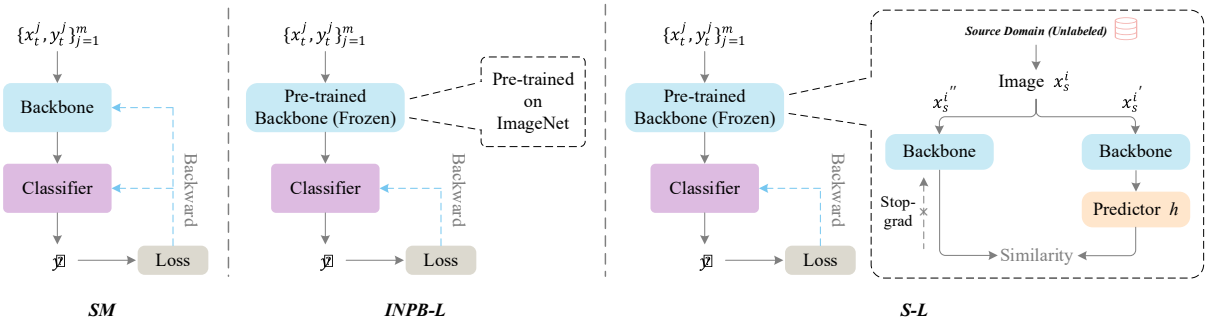

Fig. 6. The illustration of the comparison methods, including *SM*, *INPB-L*, *S-L*.

Table 3
Leaf disease identification performance for different methods. *SM*, *INPB-L*, and *S-L* are abbreviations for Supervised Method, ImageNet Pre-trained Backbone-Linear, and SimSiam-Linear, respectively. Best results are in bold.

Methods	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	Acc (%)
SM	50.58	47.76	47.70	50.37
INPB-L	63.32	63.15	62.92	63.11
S-L	70.44	70.46	69.24	70.45
CLA	**73.42**	**73.67**	**73.00**	**73.60**

Table 4
Performance results of CLA and its variants on the dataset. CLA w/o-loss_{mmd} means CLA is without maximum mean difference loss, CLA w/o-DAL means CLA is without domain adaptation layer. Best results are in bold.

	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	Acc (%)
CLA w/o-loss_{mmd}	68.77	68.87	68.26	68.80
CLA w/o-DAL	65.27	64.98	64.24	64.99
CLA	**68.93**	**68.99**	**68.42**	**68.91**

the momentum is 0.9. We refer to the paper (Chen and He, 2021; Güldenring and Nalpantidis, 2021) to set other hyperparameters, details are as follows. In addition, all models are implemented in PyTorch 1.7.0, running on two NVIDIA TITAN RTX GPUs.

CLA and its variants: CLA training framework includes two stages, i.e., pre-training and fine-tuning. In the pre-training stage, there are different models pre-trained on the source domain dataset (different volumes or different mix ratios) in different experiments. More detail about the source domain dataset is described in the experimental sections. In general, we use SGD with a learning rate = 0.03 and a cosine decay for 300 epochs, batch size = 256. In the fine-tuning stage, the DAL and classifier are trained with the data from the target domain dataset. We use SGD with a learning rate = 0.02 and a cosine decay for 300 epochs, batch size = 32. According to Equation (6), L_{MMD} and its coefficient α were applied to the loss function when DAL is used. According to Tzeng et al. (2014), α is generally set to 0.25.

Comparison methods: For *SM*, we set the learning rate = 0.03, epochs = 300, and batch size = 256 for training. For *INPB-L*, we adopted the same hyperparameter settings as CLA fine-tuning stage. For *S-L*, we configured the same hyperparameter settings as CLA in both the pre-training and fine-tuning stages.

Notably, self-supervised learning is limited by time and computing power, so it is unrealistic to extensively adjust parameters for CLA and other self-supervised learning methods, and the above-mentioned hyperparameters may not be necessarily optimal for model performance. We aim to illustrate the effectiveness of the method and the influence of various factors in this paper. Refer to Cole et al. (2022) for better models. In addition, all linear evaluation and fine-tuning here freeze the parameters of the encoder and only update the rest of the parameters, which is different from the settings of other self-supervised papers (Cole et al., 2022).

4.2. Comparison study

In comparative experiments, we train each method according to the hyperparameter settings of Section 4.1.3. For non-ImageNet pre-trained methods, 20,000 plant leaf images with a 10% mix ratio, i.e., 18,000 images from the Plantvillage dataset and 2000 images from the synthetic dataset, are used to pre-train the model. For all methods, the labeled data for fine-tuning is 760 images (0% mix ratio) in 38 classes, with 20 images in each class.

The comparative experiment results are shown in Table 3. Overall, CLA outperforms other contrastive learning methods in all respects, especially supervised learning methods by a large margin. Specifically, CLA significantly outperforms supervised methods with a small amount of labeled data and is 24.32% higher than the SM method on average on four metrics, which amply demonstrates the effectiveness of CLA with a small number of labeled samples as well as the feasibility of the unsupervised method. Comparing the results of S-L, the performance of CLA with domain adaptation is better. In addition, we can find that training based on ImageNet pre-training weights can improve the accuracy and the convergence rate of the model, which is consistent with (Güldenring and Nalpantidis, 2021). In short, we believe that the ImageNet pre-training weights provide a good initial value for the model weights to avoid falling into a locally optimal solution in model training and secure the correctness of model parameter updates, thereby enabling faster convergence and higher accuracy.

4.3. Ablation study

We verify the effectiveness of DAL and MMD loss (loss_{mmd}) in improving the domain adaptation of CLA. Specifically, according to the strategy of CLA, we utilized 10,000 plant leaf images with a 20% mix ratio (i.e., 8000 images from the Plantvillage dataset and 2000 images from the Synthetic dataset) to pre-train the model and utilized 760 labeled images which are the same as the fine-tune dataset in Section 4.2 to fine-tune the model. Additionally, we increase the training epochs to 800 and keep other hyperparameters the same as in Section 4.1.3, this is to achieve convergence to exclude other factors. As shown in Table 4, CLA w/o-loss_{mmd} represents that without loss_{mmd} but with domain adaptation layer in CLA; CLA w/o-DAL represents that without domain adaptation layer in CLA, i.e., linear evaluation.

It can be seen from Table 4 that the domain adaptation layer and its related loss terms added to improve the domain adaptation ability of the model is effective. Specifically, the addition of DAL is the key to developing the domain adaptation ability, which improves the accuracy of the CLA model by 3.92%. In addition, according to subsequent experiments, we found that the addition of DAL also slowed down the convergence speed of the model, requiring more training rounds. Simply, the addition of DAL makes more parameters need to be optimized and updated, thus requiring more training rounds. Compared with DAL, loss_{mmd} makes a small improvement of CLA accuracy to 0.11%. The α of the CLA is set to 0.25 in Table 4, which we thought was a small value of

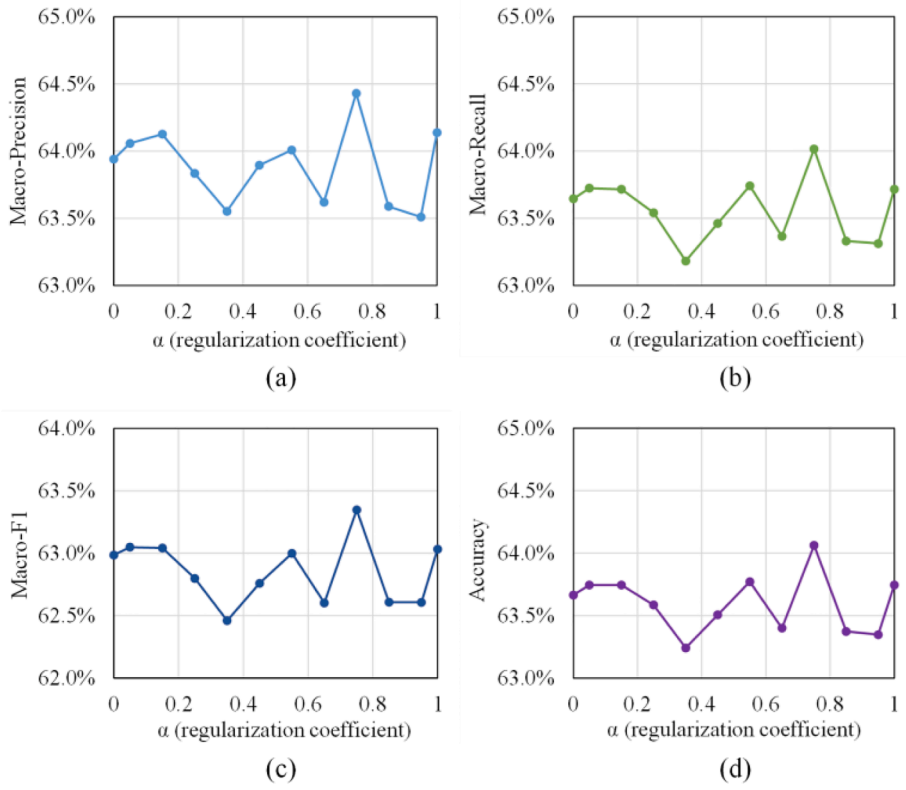

Fig. 7. Performance results of CLA with varying hyperparameters α. (a) Macro-Precision; (b) Macro-Recall; (c) Macro-F1; (d) Accuracy.

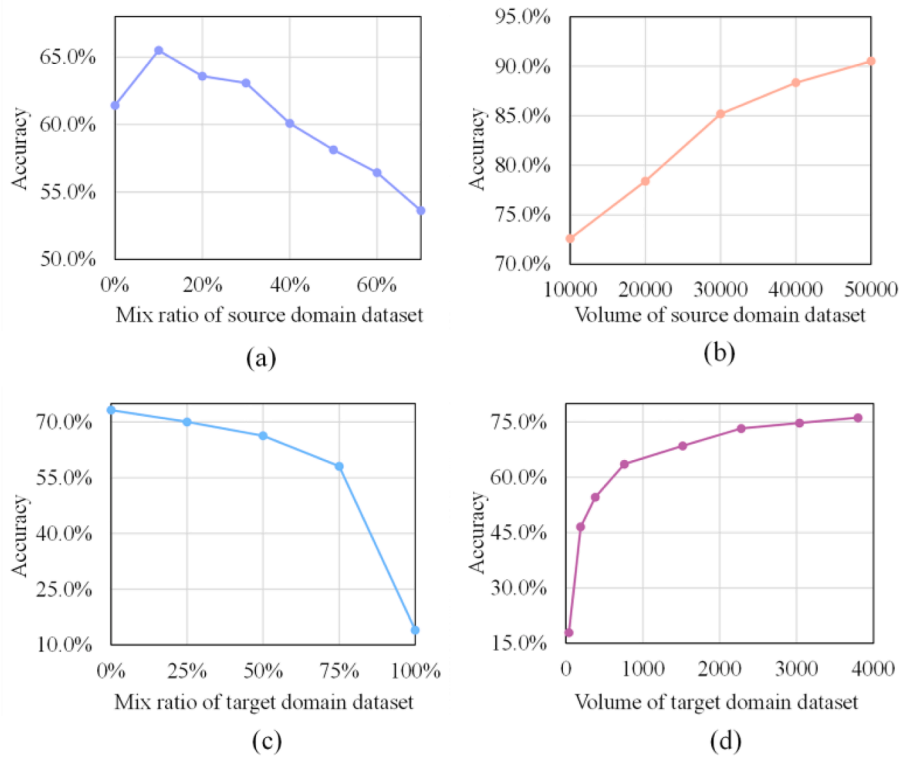

Fig. 8. Accuracy performance of CLA trained on datasets (source domain dataset, target domain dataset) with different mix ratios and volumes. (a) Source domain datasets with different mix ratios; (b) Source domain datasets with different volumes. (c) Target domain datasets with different mix ratios; (d) Target domain datasets with different volumes.

Table 5
Performance results of CLA with different numbers of classes. Best results are in bold.

Number of classes	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	Acc (%)
2	**96.93**	**96.83**	**96.87**	**96.88**
4	95.53	95.45	95.48	95.31
8	83.57	83.63	83.46	83.63
16	82.45	82.25	82.23	82.25
32	76.77	76.81	76.52	76.81
38	76.15	76.25	75.90	76.17

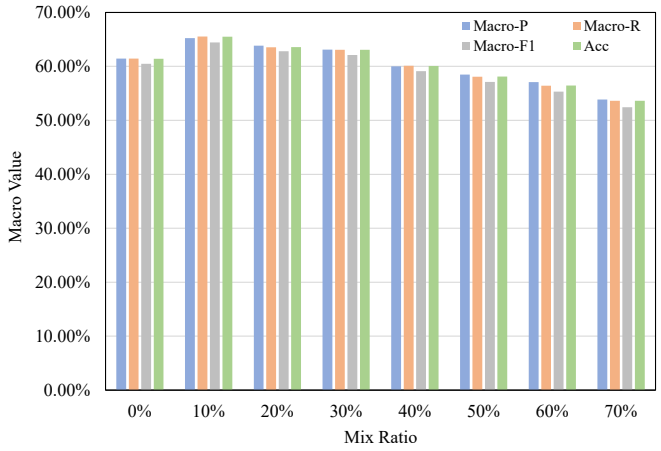

Fig. 9. Results of CLA pre-trained on D_s^{MR} with different mix ratios. D_s^{MR} is the source domain dataset (pre-training dataset) with different mix ratios. Macro value includes Macro-P, Macro-R, Macro-F1, and Accuracy.

Table 6
Performance results of CLA pre-trained on $D_s^{10\%}$ with different volumes. $D_s^{10\%}$ is the source domain dataset (pre-training dataset) with a 10% mix ratio but possessing different volumes. Best results are in bold.

Dataset Volume	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	Acc (%)
10 k	72.80	72.66	72.24	72.59
20 k	78.44	78.44	78.10	78.39
30 k	85.76	85.23	85.20	85.20
40 k	88.78	88.37	88.32	88.35
50 k	**90.64**	**90.58**	**90.40**	**90.52**

α, resulting in a limited impact of this loss term on the domain adaptation ability. Therefore, we conducted experiments on the effect of α on the model, and the results are shown in Fig. 7. We modify the number of training epochs to 300 and keep other training hyperparameters consistent with the CLA in Table 4. The result shows that setting α = 0.75 to obtain optimal performance, and α = 0.55 to obtain sub-optimal performance, the two with a difference of 0.29% in acc. Overall, increasing the value of α will make the model have better domain adaptation ability, resulting in higher accuracy performance. For α = 0 in Fig. 7, although the model performs better than some settings, we believe that this is caused by insufficient training epochs. In Table 4, we ensure enough training epochs, and the model accuracy of α = 0 (CLA w/o-$loss_{mmd}$) is lower than that of α = 0.25 (CLA). In general, in the range of 0–1, the value of α has a low impact on the accuracy of the final model, but it may be better to set a higher value of α, which is worth exploring later. Due to the difficulty of extensive parameter tuning, in order to ensure the rationality of the α setting, we set α to 0.25 in most experiments, which is consistent with Tzeng et al. (2014).

4.4. Influencing factors study

To further explore our proposed method, we conducted the influencing factors experiments. Specifically, we first explored the impact of the training datasets (source domain dataset, target domain dataset) with different mix ratios and volumes to CLA, and then studied the impact of the number of classes. Moreover, hyperparameters not described in this part are all follow Section 4.1.3. Due to the limited space, the simplified experimental results and analysis are shown in this section. See Appendix A for the detailed experimental content.

For experiments on different training datasets, we conducted four sub-experiments, including an experiment on source domain datasets with various mix ratios, an experiment on source domain datasets with various volumes, an experiment on target domain datasets with various mix ratios, and an experiment on target domain datasets with various volumes. For experiments on source domain dataset with various mix ratios, we varied mix ratios of the source domain dataset in the range of {0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%} while fixing the volume of source domain dataset, target domain dataset as {10 k, 760} images. For experiments on the target domain dataset with various mix ratios, we varied mix ratios of the target domain dataset in the range of {0%, 25%, 50%, 75%, 100%} while fixing the volume of the source domain dataset, target domain dataset as {10 k, 2280} images. As shown in Fig. 8, we only present the accuracy performance due to similar observations on other evaluation metrics. Fig. 8 (a), (c) shows the impact of mix ratios on CLA, which demonstrates that the accuracy decreases with the increase of mix ratios. This result means that the distribution shift data in training data hurts the model performance. Note that CLA benefits from the source domain dataset including a small amount of distribution-shifted data.

For experiments on various source domain dataset volumes, we varied source domain dataset volume as {10 k, 20 k, 30 k, 40 k, 50 k} images while fixing the mix ratio as 10%. Meanwhile, the same target domain dataset as Section 4.2 was utilized for fine-tuning. For

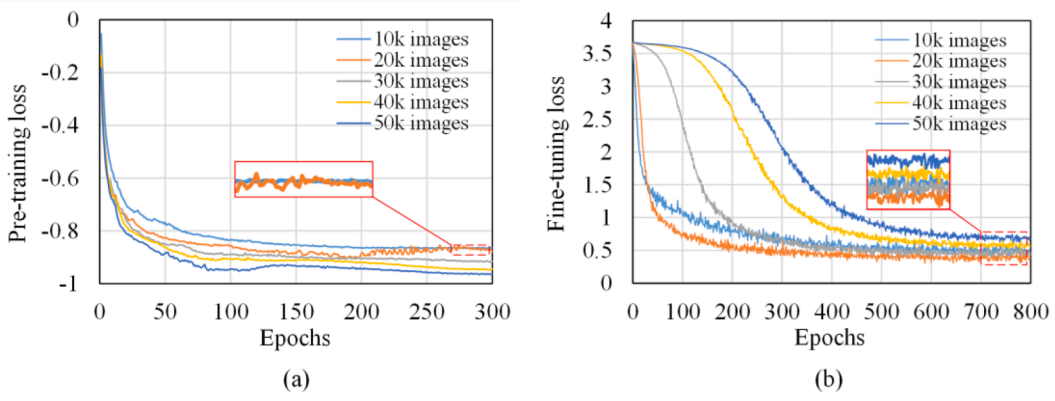

Fig. 10. Training loss of encoder (ResNet 18) pre-trained on source domain dataset with different volumes. (a) Pre-training loss curves; (b) Fine-tuning loss curves.

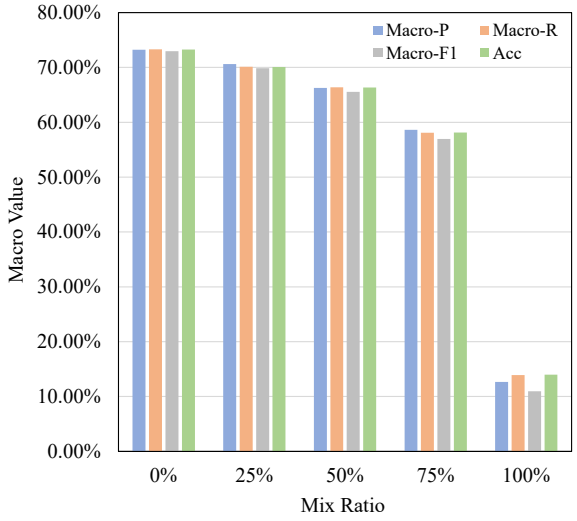

Fig. 11. Results of CLA fine-tuned on D_t^{MR} with different mix ratios. D_t^{MR} is the target domain dataset (fine-tuning dataset) with different mix ratios. Macro value includes Macro-P, Macro-R, Macro-F1 and Accuracy.

Table 7
Performance results of CLA fine-tuned on $D_t^{20\%}$ with different volumes. $D_t^{20\%}$ is the target domain dataset (fine-tuning dataset) with a 20% mix ratio but possessing different volumes. Best results are in bold.

Dataset Volume	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	Acc (%)
38 (1 image per class)	15.41	17.80	12.62	17.85
190 (5 images per class)	46.96	46.53	44.49	46.56
380 (10 images per class)	55.33	54.64	54.00	54.61
760 (20 images per class)	63.84	63.54	62.80	63.59
1520 (40 images per class)	68.47	68.58	68.00	68.54
2280 (60 images per class)	73.23	73.29	72.96	73.25
3040 (80 images per class)	74.73	74.82	74.45	74.74
3800 (100 images per class)	**76.15**	**76.25**	**75.90**	**76.17**

experiments on various target domain dataset volumes, we varied target domain dataset volume as {38, 190, 380, 760, 1520, 2280, 3040, 3800} images while fixing the mix ratio as 0%. The source domain dataset, which includes 10 k plant leaf images with 20% mix ratio, was employed for pre-training models. Fig. 8 (b), (d) show the impact of training dataset volumes on CLA. We can observe that the accuracy performance of CLA increases with the increase of training dataset volume. This indicates that larger volume datasets are needed, especially the source domain dataset.

All the above experiments are multi-classification tasks with 38 classes. There is fewer classification number in real-world scenarios, such as 2 classes, 4 classes, 8 classes, etc. Thus, we conducted experiments with different classification numbers. Specifically, we trained models with a classifier of 2, 4, 8, 16, 32, and 38 classifications according to the setting in Sections 4.1.3 and 4.2. As shown in Table 5, the experimental results demonstrate that the performance of CLA improves with the decrease in classification numbers. The classifiers with 2 or 4 classes achieve the best results with an accuracy of more than 95%.

5. Conclusions

In this work, we explore leaf disease identification in scenarios with a large amount of unlabeled distribution-shifted data and a small amount of labeled data. Aiming at the problem that the existing leaf disease identification method requires a large amount of labeled data and the distribution of unlabeled data is shifted, we propose a leaf disease identification method with domain adaptation called CLA. Specifically, CLA is based on self-supervised contrastive learning, which includes two stages, i.e., pre-training with forwarding and fine-tuning with domain adaptation. To improve the domain adaptation ability of the model, we add a domain adaptation layer before the classifier to align the distribution between the source domain data and the target domain data. In experiments, we validate the effectiveness of CLA and explore the impact of related factors on the performance of CLA. According to the experimental results, we can draw the following conclusions:

(1) The performance of CLA is better than the supervised methods in the same situation and the addition of the domain adaptation layer improves the domain adaptation ability of CLA;
(2) The addition of a small part of the data with distribution shifts in the source domain data set is beneficial to improving the accuracy of CLA. When $MR > 30\%$, the accuracy of CLA decreases with the increase of MR. The larger the volume of the source domain dataset and the target domain dataset, the better the performance of the model;
(3) The number of classifications of the model should not be too much.

Table 8
Performance results of CLA with different numbers of classes. Best results are in bold.

Number of classes	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	Acc (%)
2	**96.93**	**96.83**	**96.87**	**96.88**
4	95.53	95.45	95.48	95.31
8	83.57	83.63	83.46	83.63
16	82.45	82.25	82.23	82.25
32	76.77	76.81	76.52	76.81
38	76.15	76.25	75.90	76.17

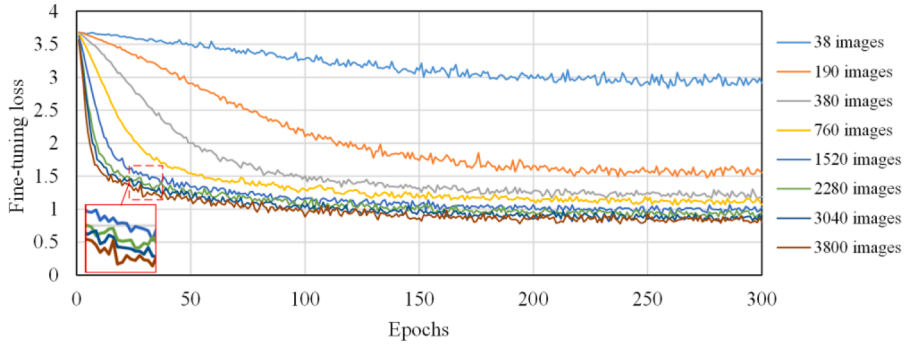

Fig. 12. Fine-tuning loss on target domain dataset with different volumes.

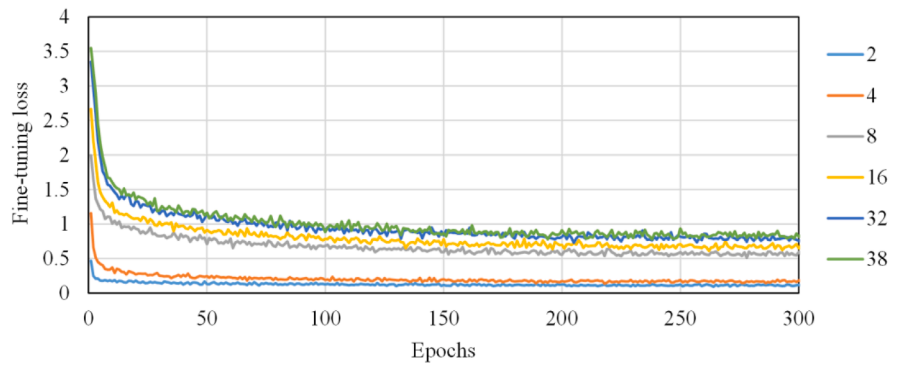

Fig. 13. Fine-tuning loss for models with different number of classes.

Based on the conclusions mentioned above, CLA can effectively deal with the problem of leaf disease identification under the scarcity of labeled data and unlabeled data with distribution shifts.

Funding

This work was supported by 2022 Guangdong Provincial Science and Technology Innovation Strategy Special Fund [pdjh2022 b0808]; Guangdong Basic and Applied Basic Research Foundation [2021A1515110554]; China Postdoctoral Science Foundation [Grant No. 2022 M721201]; The open competition program of top ten critical priorities of Agricultural Science and Technology Innovation for the 14th Five-Year Plan of Guangdong Province [2022SDZG03]; 2021 Provincial Higher Vocational Education Teaching Reform Research and Practice Project [GDJG2021068]; and Research on detection method of plant leaf disease based on self-supervised contrastive learning [YJZD2022-07].

CRediT authorship contribution statement

Ruzhun Zhao: Conceptualization, Investigation, Validation, Methodology, Writing – original draft, Resources. **Yuchang Zhu:** Investigation, Data curation, Writing – review & editing, Software. **Yuanhong Li:** Supervision, Resources.

Table 9
Detailed information of the dataset used in the experiments.

	Plantvillage	synthetic dataset	
		PlantDoc	Augmented leaf disease dataset
Apple Apple scab	630	93	300
Apple Black rot	621	0	84
Apple Cedar apple rust	275	88	216
Apple healthy	1645	91	64
Blueberry healthy	1502	115	0
Cherry (including sour) healthy	854	57	72
Cherry (including sour) Powdery mildew	1052	0	72
Corn (maize) Cercospora leaf spot Gray leaf spot	513	68	200
Corn (maize) Common rust	1192	116	132
Corn (maize) healthy	1162	0	199
Corn (maize) Northern Leaf Blight	985	191	204
Grape Black rot	1180	64	72
Grape Esca (Black Measles)	1383	0	124
Grape healthy	423	69	268
Grape Leaf blight (Isariopsis Leaf Spot)	1076	0	40
Orange Haunglongbing (Citrus greening)	5507	0	92
Peach Bacterial spot	2297	0	0
Peach healthy	360	111	0
Pepper, bell Bacterial spot	997	71	172
Pepper, bell healthy	1478	61	156
Potato Early blight	1000	116	160
Potato healthy	152	0	172
Potato Late blight	1000	105	176
Raspberry healthy	371	119	0
Soybean healthy	5090	65	0
Squash Powdery mildew	1835	130	0
Strawberry healthy	456	96	112
Strawberry Leaf scorch	1109	0	216
Tomato Bacterial spot	2127	110	112
Tomato Early blight	1000	88	176
Tomato healthy	1591	63	140
Tomato Late blight	1909	111	220
Tomato Leaf Mold	952	91	156
Tomato Septoria leaf spot	1771	151	160
Tomato Spider mites Two-spotted spider mite	1676	0	124
Tomato Target Spot	1404	0	56
Tomato mosaic virus	371	54	144
Tomato Yellow Leaf Curl Virus	5357	76	236
Total	54,303	2570	4827

Declaration of Competing Interest

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

Data availability

The authors are unable or have chosen not to specify which data has been used.

Appendix

A. Detailed experimental results for influencing factors study

To better understand the performance of CLA, we explore the impact of the training dataset on the model from the two stages (i.e., pre-training and fine-tuning). The main impact factors considered are the component and size of the training dataset, see Sections a) and b) for details. In addition, we also explore the effect of the number of classes of the classifier on CLA, see Section c) for details.

a) Impact of pre-training

i. Mix ratio of source domain dataset

To explore the effect of the source domain dataset component on CLA, we constructed several source domain datasets with different mix ratios. We denote source domain datasets with different mix ratios by D_s^{MR}, where $MR = \{0\%, 10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%\}$ is the mix ratio of the dataset. It is worth noting that due to the limitation of the number of synthetic dataset images, the mix ratio can only reach a maximum of 70%. According to the strategy of CLA, we utilised the source domain dataset with a total of 10 k images for pre-training, and then utilised the same target domain dataset as Section 4.2 for fine-tuning. In addition, other training hyperparameters remained the same as those in Section 4.1.3.

The performance of the models trained under D_s^{MR} with different mix ratios is shown in Fig. 9. We can see that, except when $MR = 0\%$, the accuracy of CLA deteriorates as the mix ratio increases. Specifically, when $MR = 10\%$, the best accuracy performance of CLA occurs and reaches 65.49%. When $MR = 70\%$, the worst accuracy performance of CLA occurs and reaches 53.63%. We believe that if the mix ratio continues to rise, the accuracy of CLA will continue to decline. It is worth noting that the performance of CLA was not the best when $MR = 0\%$. There were two main reasons: First, a small part ($MR \leqslant 30\%$) of images with complex backgrounds and different shooting angles were added to the source domain dataset, which had no influence on the semantic content of the original source domain dataset. The only change is to expand the distribution of source domain data, so that the model can see a wider data distribution during training. Secondly, due to the addition of the domain adaptation layer, CLA is more suitable for processing data with distribution shifts. For the problem of CLA accuracy degradation caused by the increase in mix ratio, we think it is worth exploring the future. At present, there have been studies on cross-domain self-supervised learning with negative samples. Cole et al. (2022) argued that pre-training on cross-domain data makes the difference between positive and negative samples large, resulting in easier pretext tasks. If the pretext task is too simple, it will lead to a decrease in the accuracy of the model. However, CLA is a self-supervised learning method without negative sample pairs, and the same phenomenon occurs. Tian et al. (2021) studied the learning mechanism of the self-supervised method without negative sample pairs, but it was still unable to answer the reason for that phenomenon. In a word, from the experiments in Fig. 9, adding a small part of distribution-shifted data in the source domain data set is beneficial to improve the accuracy of CLA, but as the mix ratio increases, the accuracy of CLA will gradually decrease.

ii. Volume of source domain dataset

To verify the effect of the size of the source domain dataset on the performance of CLA, we pre-trained the model using the source domain dataset with $MR = 10\%$, and utilised the same dataset as Section 4.2 for fine-tuning. Specifically, we denote source domain datasets of different sizes by $D_s^{10\%} = \{x_s^i\}_{i=1}^n$, where $n = \{10k, 20k, 30k, 40k, 50k\}$ is the number of images in the dataset. To make sure the model converges, we used SGD with base $lr = 400$ and a pre-training schedule for 800 epochs. In addition, other hyperparameters are consistent with those in Section 4.1.3.

As shown in Table 6, fine-tuning on the same target domain dataset, CLA performed better with the increase in the size of the source domain dataset. Even if $n = 10k$, the accuracy of CLA still reached 72.59%, and that was not the optimal result under this condition. It is worth noting that in the process of the source domain dataset size increase from 10 k → 20 k → 30 k, the improvement of CLA in accuracy was stable at about 6%, while the improvement of CLA was only 3.15% and 2.17% when the dataset size is changed from 30 k → 40 k → 50 k. As shown in Fig. 10(a), the losses of models pre-trained on 50 k and 40 k images were smaller than that of the other three models. That phenomenon showed that the model pre-trained on 50 k and 40 k images can obtain better visual representation. As a result, better visual representation made the classifier perform better, i.e. higher accuracy. As shown in Fig. 10(b), in the fine-tuning stage, the loss of models pre-trained on the 50 k and 40 k images decreased more slowly than the loss of other models. These two loss curves still had a decreasing trend until 800 epochs were completed. Therefore, increasing the training epochs, models pre-trained on 40 k and 50 k images had the potential to be higher-accuracy models. That is why the accuracy gap of CLA is relatively small when the source domain dataset size is changed from 30 k → 40 k → 50 k. Increasing the volume of the source domain dataset significantly improves the accuracy of CLA, and it is not difficult to increase the source domain dataset size in real-world scenarios. In addition, Cole et al. (2022) showed that it is a little benefit when the pre-training dataset size is more than 500 k images. Fortunately, the pre-training images in our experiments were only at most 50 k, which was much less than the 500 k images. Therefore, we can significantly improve the volume of the source domain dataset and the performance of CLA will be further improved. In general, increasing the volume of the source domain dataset is very helpful to improve the accuracy of CLA, while it is a simple task to increase the source domain dataset.

b) Impact of fine-tuning

i. Mix ratio of target domain dataset

Similar to the experimental setup in Section a), we constructed five target domain datasets with different mix ratios to fine-tune the model. We denote the different target domain datasets by D_t^{MR}, where $MR = \{0\%, 25\%, 50\%, 75\%, 100\%\}$. Specifically, we utilised the source domain dataset with a total number of 10 k images with a 20% mix ratio for pre-training, and then used the target domain dataset with a total number of 2,280 images (38 classes, 60 images per class.) for fine-tuning. In addition, other training hyperparameters remained the same as those in Section 4.1.3.

As shown in Fig. 11, as the mix ratio of the target domain dataset increases, the accuracy of CLA gradually decreases. When $MR = 100\%$, the accuracy of CLA showed a cliff-like decline, which was only 13.98%. That accuracy was 44.15% lower than the accuracy of $MR = 75\%$. When $MR = 0\%$, CLA had the highest accuracy which reached 73.25%. In terms of the overall trend, the decrease in CLA accuracy caused by the increase in the mix ratio of the target domain dataset is similar to that of Section a), but when the mix ratio of the target domain dataset is relatively small, the accuracy of CLA does not improve. The target domain dataset worked in the fine-tuning stage, and the fine-tune stage of CLA mainly trained the classifier, so when the data with distribution shift was added to the training of the fine-tuning stage, it led to the decision boundary of the classifier is shifted. As a result, the trained model made more misclassifications. Therefore, as long as the distribution-shifted data is mixed in the target domain dataset, the performance of the model will be degraded. Although the domain adaptation ability of CLA is proved in Sections 4.2 and 4.3, when $MR = 100\%$, the performance of CLA still falls off a cliff, mainly because the target domain dataset cannot provide data without distribution shifts. Thus, the domain adaptation layer cannot work, resulting in the loss of the domain adaptation capability of CLA. When using CLA for plant leaf disease identification, it is best to provide a certain amount of distributed unbiased data to fine-tune the model. In general, the accuracy of CLA will gradually decrease with the increase of the mix ratio, and there will be a cliff-like decrease when $MR = 100\%$.

ii. Volume of target domain dataset

In order to verify the impact of target domain dataset size on the performance of CLA, we pre-trained the model using the source domain dataset which includes 10,000 plant leaf images with $MR = 20\%$, and fine-tuned the model on target domain datasets of different sizes with $MR = 0\%$. Specifically, we denote the source domain datasets of different sizes by $D_t^{0\%} = \{x_t^j\}_{j=1}^{m}$, where $m = \{38, 190, 380, 760, 1520, 2280, 3040, 3800\}$ is the number of images in the dataset, corresponding to 1/5/10/20/40/60/80/100 images of per class, for a total of 38 categories. In addition, other training hyperparameters remained the same as those in Section i.

As shown in Table 7, as the dataset size increases, the accuracy of CLA is gradually improved. When $m = 38$, CLA performed the worst with an accuracy of only 17.85% and an F1-score of 12.62%. When $m = 3800$, CLA performed the best with an accuracy of 76.17% and an F1-score of 75.90%. The difference between the best and worst performance of CLA was 58.32% in accuracy and 63.28% in F1-score. Obviously, when the dataset increases, the train steps in each epochs increases, i.e. the number of iterative updates of the model parameters increases. Thus, the model fitted the training data better due to more training steps. In addition, the increase in the target domain dataset also meant that the data covered was more widely distributed, so the model saw more data in the fine-tuning stage, which improved the performance of the model. However, according to the above experimental settings, the epochs required for model training to converge were different under different dataset sizes, so the statistics in Table 7 were not the best results.

As shown in Fig. 12, there are subtle differences in the loss curves of CLA fine-tuning in target domain datasets with different sizes. As the dataset size grows, the loss decreases faster and the model converges faster, mainly because the training steps in each epoch increase as the dataset size grows. When $m = 38$, the loss of the model dropped slowly and still had a downward trend after finishing training epochs. If the training epochs of the model are appropriately increased, the performance of the model will be better, but it may bring about the problem of over-fitting. The problem of over-fitting in the case of small sample training is also a concern in few-shot learning. More relevant content about over-fitting can be obtained from the previous works (Wang et al., 2020; Zhang et al., 2022b). When $m > 760$, the difference in fine-tuning loss after CLA training reached convergence was small, and the convergence speed was almost the same. However, from the enlarged image in the lower left corner, it can be seen that the larger the target domain dataset, the faster the model achieves convergence and the faster it converges. In addition, if the target domain dataset was larger, the loss after model convergence was smaller. In terms of real-world application, we do not need to use thousands of images for fine-tuning, because the benefit of adding images after the target domain dataset exceeds 760 images is relatively small, and adding image annotations is a time-consuming task. For this part of the experiments, we believe that using 760 labeled images for fine-tuning can achieve the trade-off between model performance and computational cost, which is the reason that most of the experiments in this work use 760 images for fine-tuning.

c) Impact of the number of classes

All the previous experiments are leaf disease classification for 38 classes, but for the application of real-world scenarios, the classifier doesn't need to have so many classification classes. Thus, we explored the effect of the number of classifications of the classifier on the performance of CLA. To avoid the impact of datasets on CLA, we unified the total number of images of the source domain dataset to 10 k with a 20% mix ratio and the total number of images of the target domain dataset to 3800 with a 0% mix ratio. In addition, the hyperparameter for pre-training and fine-tuning are consistent with those in Section 4.1.3. Based on the aforementioned settings, we trained models with a classifier of 2, 4, 8, 16, 32, and 38 classifications and tested the accuracy of these models.

As shown in Table 8, as the number of classifier classifications increases, the accuracy of CLA also decreased. When the number of classifications is 2 or 4, the accuracy of CLA can exceed 95%, which is also the accuracy that many existing supervised methods can achieve, but the difference is that these supervised methods require a large amount of labeled data for training. In short, when the number of classifications of the classifier is smaller, the classification boundary that the classifier needs to determine is simpler, so the classifier can easily achieve a relatively high accuracy. In addition, we also found that: when the number of classifications was changing as $4 \rightarrow 8$, $16 \rightarrow 32$, the accuracy of CLA would drop sharply, dropping by 11.68% and 5.44% respectively. When the number of classifications was changed as $2 \rightarrow 4$, $8 \rightarrow 16$, the accuracy of CLA decreased by a small margin, by 1.57% and 1.38%, respectively. Fig. 13 describes the fine-tuning loss of the model with different number of classes, we can observe the same phenomenon mentioned above from the distance of different loss curves in the vertical direction, which shows that the phenomenon of accuracy variation we

observed in Table 8 is not accidental. We are not yet able to answer the reasons for this phenomenon. Fortunately, this phenomenon provided us with a clear direction for applying CLA to real-world scenarios, allowing us to double the number of model classifications without a dramatic decrease in accuracy.

B. Datasets detail

References

Annabel, L.S.P., Annapoorani, T., Deepalakshmi, P., 2019. Machine learning for plant leaf disease detection and classification–a review. Int. Conf. Commun. Signal Process. (ICCSP) 2019, 538–542.

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M.G., Bereciartua, A., Alvarez-Gila, A., 2020. Few-Shot Learning approach for plant disease classification using images taken in the field. Comput. Electron. Agric. 175, 105542.

Ashwinkumar, S., Rajagopal, S., Manimaran, V., Jegajothi, B., 2022. Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. Mater. Today:. Proc. 51, 480–487.

Bansal, P., Kumar, R., Kumar, S., 2021. Disease detection in apple leaves using deep convolutional neural network. Agriculture 11 (7), 617.

Bommanapally, V., Ashaduzzman, M., Malshe, M., Chundi, P., Subramaniam, M., 2021. Self-supervised learning approach to detect corrosion products in biofilm images. IEEE Int. Conf. Bioinform. Biomed. (BIBM) 2021, 3555–3561.

Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.-P., Schölkopf, B., Smola, A.J., 2006. Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics 22 (14), e49–e57.

Breiki, F. al, Ridzuan, M., Grandhe, R., 2021. Self-Supervised Learning for Fine-Grained Image Classification. ArXiv Preprint ArXiv:2107.13973.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A., 2020. Unsupervised learning of visual features by contrasting cluster assignments. Adv. Neural Inf. Proces. Syst. 33, 9912–9924.

Chen, J., Chen, J., Zhang, D., Sun, Y., Nanehkaran, Y.A., 2020a. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agric. 173, 105393.

Chen, L., Cui, X., Li, W., 2021. Meta-learning for few-shot plant disease detection. Foods 10 (10), 2441.

Chen, X., He, K., 2021. Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15750–15758.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020b. A simple framework for contrastive learning of visual representations. Int. Conf. Mach. Learn. 1597–1607.

Cole, E., Yang, X., Wilber, K., mac Aodha, O., Belongie, S., 2022. When does contrastive visual representation learning work? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 14755–14764.

Dwivedi, R., Dey, S., Chakraborty, C., Tiwari, S., 2021. Grape disease detection network based on multi-task learning and attention features. IEEE Sens. J. 21 (16), 17573–17580.

Falaschetti, L., Manoni, L., Rivera, R.C.F., Pau, D., Romanazzi, G., Silvestroni, O., Tomaselli, V., Turchetti, C., 2021. A low-cost, low-power and real-time image detector for grape leaf esca disease based on a compressed CNN. IEEE J. Emerging Sel. Top. Circuits Syst. 11 (3), 468–481.

Fang, U., Li, J., Lu, X., Gao, L., Ali, M., Xiang, Y., 2021. Self-supervised cross-iterative clustering for unlabeled plant disease images. Neurocomputing 456, 36–48.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al., 2020. Bootstrap your own latent-a new approach to self-supervised learning. Adv. Neural Inf. Proces. Syst. 33, 21271–21284.

Güldenring, R., Nalpantidis, L., 2021. Self-supervised contrastive learning on agricultural images. Comput. Electron. Agric. 191, 106510.

Gupta, S., Verma, S., Thakur, R., 2019. Phytosanitary requirement for import of horticulture crops. Int. J. Curr. Microbiol. App. Sci 8 (2), 2871–2886.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738.

Hee-Jin, Y., Chang-Hwan, S., n.d. Leaf spot attention network for apple leaf disease identification.

Hewarathna, I., 2022. Augmented leaf disease dataset.

Hu, Y., Zhan, J., Zhou, G., Chen, A., Cai, W., Guo, K., Hu, Y., Li, L., 2022. Fast forest fire smoke detection using MVMNet. Knowl.-Based Syst. 241 https://doi.org/10.1016/j.knosys.2022.108219.

Hughes, D., Salathé, M., others, 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. ArXiv Preprint ArXiv:1511.08060.

Jadon, S., 2020. SSM-net for plants disease identification in low data regime. In: 2020 IEEE/ITU International Conference on Artificial Intelligence for Good (AI4G), pp. 158–163.

Ji, M., Zhang, L., Wu, Q., 2020. Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks. Inform. Process. Agric. 7 (3), 418–426.

Jogekar, R.N., Tiwari, N., 2021. A review of deep learning techniques for identification and diagnosis of plant leaf disease. Smart Trends Comput. Commun.: Proc. SmartCom 2020, 435–441.

Kar, S., Nagasubramanian, K., Elango, D., Nair, A., Mueller, D.S., O'Neal, M.E., Singh, A. K., Sarkar, S., Ganapathysubramanian, B., Singh, A., 2021. Self-Supervised Learning Improves Agricultural Pest Classification. AI for Agriculture and Food Systems.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2323. https://doi.org/10.1109/5.726791.

Li, Y., Chao, X., 2021. Semi-supervised few-shot learning approach for plant diseases recognition. Plant Methods 17 (1), 1–10.

Li, Y., Yang, J., 2021. Meta-learning baselines and database for few-shot classification in agriculture. Comput. Electron. Agric. 182, 106055.

Li, L., Zhang, S., Wang, B., 2021. Plant disease detection and classification by deep learning—a review. IEEE Access 9, 56683–56698.

Lu, J., Tan, L., Jiang, H., 2021. Review on convolutional neural network (CNN) applied to plant leaf disease classification. In Agriculture (Switzerland) (Vol. 11, Issue 8). MDPI AG. https://doi.org/10.3390/agriculture11080707.

Lu, X., Yang, R., Zhou, J., Jiao, J., Liu, F., Liu, Y., Su, B., Gu, P., 2022. A hybrid model of ghost-convolution enlightened transformer for effective diagnosis of grape leaf disease and pest. J. King Saud Univ.-Comput. Inform. Sci. 34 (5), 1755–1767.

Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7, 1419.

Najafian, K., Ghanbari, A., Stavness, I., Jin, L., Shirdel, G. H., Maleki, F., 2021. A semi-self-supervised learning approach for wheat head detection using extremely small number of labeled samples. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 1342–1351.

Ngugi, L.C., Abelwahab, M., Abo-Zahhad, M., 2021. Recent advances in image processing techniques for automated leaf pest and disease recognition–A review. Inform. Process. Agric. 8 (1), 27–51.

Nuthalapati, S.V., Tunga, A., 2021. Multi-domain few-shot learning and dataset for agricultural applications. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 1399–1408.

Rayhan, Y., Setyohadi, D.B., 2021. Classification of Grape Leaf Disease Using Convolutional Neural Network (CNN) with Pre-Trained Model VGG16. In: 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), 1–5.

Shantkumari, M., Uma, S.V., 2021. Grape leaf segmentation for disease identification through adaptive Snake algorithm model. Multimed. Tools Appl. 80 (6), 8861–8879.

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., & Batra, N., 2020. PlantDoc: a dataset for visual plant disease detection. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD (pp. 249–253).

Tang, Z., Yang, J., Li, Z., Qi, F., 2020. Grape disease image classification based on lightweight convolution neural networks and channelwise attention. Comput. Electron. Agric. 178, 105735.

Tian, Y., Chen, X., Ganguli, S., 2021. Understanding self-supervised learning dynamics without contrastive pairs. International Conference on Machine Learning 10268–10278.

Too, E.C., Yujian, L., Njuki, S., Yingchun, L., 2019. A comparative study of fine-tuning deep learning models for plant disease identification. Comput. Electron. Agric. 161, 272–279.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T., 2014. Deep domain confusion: Maximizing for domain invariance. *ArXiv Preprint ArXiv:1412.3474*.

Verma, S., Chug, A., Singh, A.P., 2020. Impact of hyperparameter tuning on deep learning based estimation of disease severity in grape plant. International Conference on Soft Computing and Data Mining 161–171.

Wang, Y., Wang, S., 2021. IMAL: An Improved Meta-learning Approach for Few-shot Classification of Plant Diseases. In: 2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE), 1–7.

Wang, W., Zhang, J., Cao, Y., Shen, Y., Tao, D., 2022b. Towards Data-Efficient Detection Transformers. https://github.com/encounter1997/DE-DETRs.

Wang, F., Rao, Y., Luo, Q., Jin, X., Jiang, Z., Zhang, W., Li, S., 2022a. Practical cucumber leaf disease recognition using improved Swin Transformer and small sample size. Comput. Electron. Agric. 199 https://doi.org/10.1016/j.compag.2022.107163.

Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M., 2020. Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys (Csur) 53 (3), 1–34.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., & Shao, L., 2021. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. https://github.com/whai362/PVT.

Wen, H., Hu, X., Peng, Y., 2021. Self-supervised Domain Adaptation for Satellite Imagery Classification via Representation Learning. In: 2021 IEEE 2nd International

Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 464–469.
Yan, K., Guo, X., Ji, Z., Zhou, X., 2021. Deep Transfer Learning for Cross-Species Plant Disease Diagnosis Adapting Mixed Subdomains. IEEE/ACM Transactions on Computational Biology and. Bioinformatics.
Yang, J., Guo, X., Li, Y., Marinello, F., Ercisli, S., Zhang, Z., 2022. A survey of few-shot learning in smart agriculture: developments, applications, and challenges. Plant Methods 18 (1), 1–12.
Yang, R., Lu, X., Huang, J., Zhou, J., Jiao, J., Liu, Y., Liu, F., Su, B., Gu, P., 2021. A Multi-Source Data Fusion Decision-Making Method for Disease and Pest Detection of Grape Foliage Based on ShuffleNet V2. Remote Sens. (Basel) 13 (24), 5102.
Zhang, Y., Wa, S., Liu, Y., Zhou, X., Sun, P., Ma, Q., 2021. High-Accuracy Detection of Maize Leaf Diseases CNN Based on Multi-Pathway Activation Function Module. Remote Sens. (Basel) 13 (21), 4218.
Zhang, Y., Zheng, Y., Xu, X., Wang, J., 2022b. How Well Do Self-Supervised Methods Perform in Cross-Domain Few-Shot Learning? ArXiv Preprint. ArXiv:2202.09014.
Zhang, L., Zhou, G., Lu, C., Chen, A., Wang, Y., Li, L., Cai, W., 2022a. MMDGAN: A fusion data augmentation method for tomato-leaf disease identification. In: Applied Soft Computing, Vol. 123. Elsevier Ltd. https://doi.org/10.1016/j.asoc.2022.108969
Zhao, Y., Chen, Z., Gao, X., Song, W., Xiong, Q., Hu, J., Zhang, Z., 2021. Plant disease detection using generated leaves based on doubleGAN. IEEE/ACM Transactions on Computational Biology and. Bioinformatics.
Zhou, C., Zhang, Z., Zhou, S., Xing, J., Wu, Q., Song, J., 2021. Grape leaf spot identification under limited samples by fine grained-GAN. IEEE Access 9, 100480–100489.
Zinonos, Z., Gkelios, S., Khalifeh, A.F., Hadjimitsis, D.G., Boutalis, Y.S., Chatzichristofis, S.A., 2021. Grape leaf diseases identification system using convolutional neural networks and LoRa technology. IEEE Access 10, 122–133.