agronomy

MDPI

Article

HMFN-FSL: Heterogeneous Metric Fusion Network-Based Few-Shot Learning for Crop Disease Recognition

Wenbo Yan [1], Quan Feng [1,*], Sen Yang [1,*], Jianhua Zhang [2] and Wanxia Yang [1]

[1] College of Mechanical and Electrical Engineering, Gansu Agricultural University, Lanzhou 730070, China; yanwb@st.gsau.edu.cn (W.Y.); yangwanxia@gsau.edu.cn (W.Y.)

[2] Agricultural Information Institute, Chinese Academy of Agricultural Sciences, Beijing 100081, China; zhangjianhua@caas.cn

* Correspondence: fquan@gsau.edu.cn (Q.F.); yangsen@gsau.edu.cn (S.Y.)

Abstract: The high performance of deep learning networks relies mainly on massive data. However, collecting enough samples of crop disease is impractical, which significantly limits the intelligent diagnosis of diseases. In this study, we propose Heterogeneous Metric Fusion Network-based Few-Shot Learning (HMFN-FSL), which aims to recognize crop diseases with unseen categories using only a small number of labeled samples. Specifically, CBAM (Convolutional Block Attention Module) was embedded in the feature encoders to improve the feature representation capability. Second, an improved few-shot learning network, namely HMFN-FSL, was built by fusing three metric networks (Prototypical Network, Matching Network, and DeepEMD (Differentiable Earth Mover's Distance)) under the framework of meta-learning, which solves the problem of the insufficient accuracy of a single metric model. Finally, pre-training and meta-training strategies were optimized to improve the ability to generalize to new tasks in meta-testing. In this study, two datasets named Plantvillage and Field-PV (covering 38 categories of 14 crops and containing 50,403 and 665 images, respectively) are used for extensive comparison and ablation experiments. The results show that the HMFN-FSL proposed in this study outperforms the original metric networks and other state-of-the-art FSL methods. HMFN-FSL achieves 91.21% and 98.29% accuracy for crop disease recognition on 5way-1shot, 5way-5shot tasks on the Plantvillage dataset. The accuracy is improved by 14.86% and 3.96%, respectively, compared to the state-of-the-art method (DeepEMD) in past work. Furthermore, HMFN-FSL was still robust on the field scenes dataset (Field-PV), with average recognition accuracies of 73.80% and 85.86% on 5way-1shot, 5way-5shot tasks, respectively. In addition, domain variation and fine granularity directly affect the performance of the model. In conclusion, the few-shot method proposed in this study for crop disease recognition not only has superior performance in laboratory scenes but is also still effective in field scenes. Our results outperform the existing related works. This study provided technical references for subsequent few-shot disease recognition in complex environments in field environments.

Keywords: few-shot learning; metric learning; multi-model fusion; attention; plant protection; crop disease recognition

check for updates

Citation: Yan, W.; Feng, Q.; Yang, S.; Zhang, J.; Yang, W. HMFN-FSL: Heterogeneous Metric Fusion Network-Based Few-Shot Learning for Crop Disease Recognition. *Agronomy* **2023**, *13*, 2876. https://doi.org/10.3390/agronomy13122876

Academic Editor: Roberto Marani

Received: 18 October 2023
Revised: 14 November 2023
Accepted: 21 November 2023
Published: 23 November 2023

1. Introduction

In agricultural production, timely diagnosis of crop diseases is critical to improving crop yields [1,2]. The diagnosis of crop leaf diseases is a crucial part of precision agriculture. Currently, disease diagnosis largely depends on experienced farmers or pest experts using manual observation. This method is inefficient and difficult to implement on a large scale, especially for smallholders in remote areas. With the rapid development of deep learning in the field of automatic crop leaf disease recognition, intelligent diagnosis has become feasible and has been successfully applied to various crops such as oil tea camellia, rice, tomato, cucumber, maize, citrus, and sunflowers [3–9]. However, there are still some pressing issues to be addressed in such methods. First, the long-tail distribution of disease samples may result in better performance of the model for common categories and worse performance for rare categories. Second, collecting sufficient samples of crop diseases is challenging. In addition, the annotation of the dataset requires the participation of a large number of experts in the field of agricultural diseases, which also increases the difficulty of the dataset construction. Therefore, research into a deep convolutional model that is suitable for learning diseases with small samples is crucial to solving the problem of insufficient disease data.

Currently, there are usually two ways to alleviate the problem caused by data shortage. Data augmentation increases the amount of data through operations such as scaling and rotating images or synthesizing sample data using Generative Adversarial Networks (GANs). For example, Hu et al. [10] augmented disease spot images using the C-DCGAN method and achieved an average accuracy of 90.00% for the recognition of tea diseases. Chen et al. [11] also solved the problem of insufficient samples and achieved an accuracy of 97.78% in apple disease classification tasks by using the CycleGAN network to generate synthetic samples. Cap et al. [12] proposed an image transformation system with its own mechanism (LeafGAN) for crop leaf disease features, which achieved better image generation than CycleGAN. Data augmentation can improve the learning of rare classes by generating more samples to balance the training data. However, for small datasets, such methods may result in the information learnt by the model being too similar, thus limiting its ability to generalize. Transfer-learning-based methods are first pre-trained on a source dataset to obtain a generic feature representation, and then the network is fine-tuned using a small amount of target data. For example, Gulazer et al. [13] used a migration learning strategy trained on a CNN network to classify seeds and achieved 99% accuracy on a test set containing only 234 images. Mamat et al. [14] used the YOLO model developed based on migration learning and successfully achieved high performance recognition of oleaginous palm trees. Zhang et al. [15] developed a lotus leaf pest and disease recognition model using transfer learning after improving DenseNet based on the Plantvillage dataset for transfer learning to recognize lotus-leaf-related diseases and achieved an accuracy of 91.34%. Li et al. [16] used DenseNet on Plantvillage for pre-training and the tea dataset for fine-tuning. This resulted in 92.66% accuracy for tea disease recognition with insufficient samples. Yang et al.'s study [17] achieved 97.23% accuracy in corn disease recognition based on MobileNetV2 for transfer learning. In addition, Gulazer [18] improved MobileNetV2 by obtaining TL-MobileNetV2 and training it using a migration strategy for MobileNetV2, obtaining extremely high performance on a fruit classification task. Although transfer learning has alleviated the sample shortage problem in crop leaf disease recognition to some extent, these methods still have some limitations. First, the model only predicts well for disease categories in the training samples but cannot generalize to untrained disease categories. Second, it is also difficult to achieve reliable performance with transfer learning when the number of available samples is minimal (e.g., one sample) because too-sparse samples do not provide enough information to support model training. Therefore, the core challenge for small-sample disease recognition is to rely on only a few samples to learn and make the model achieve the generalization ability. However, solving these challenges requires innovations in the models and algorithms themselves, not just operating at the data level.

In recent years, few-shot learning (FSL) based on meta-learning has provided new ideas for the recognition of foliar diseases in crops with small sample sizes. Metric learning-based methods are an effective classification method in FSL. These methods utilize human-summarized metric functions [19], such as Euclidean distance, cosine distance, and other non-parametric or less parametric modules, for classification instead of traditional linear layers. This model fine-tunes the feature encoder to ensure that samples from the same class are positioned close together in the measurement space. In contrast, samples from different classes are kept well apart. The metric learning method avoids parameter learning within the linear layer, while allowing the model to generalize to novel categories. At present, FSL based on metric learning has a promising application in crop leaf disease recognition. Pan et al. [20] proposed a Siamese network-based FSL learning method for recognition of crop leaf diseases. For Monocotyledonous crops, 68.57% and 76.95% accuracies were achieved on 5way-5shot and 10way-10shot, respectively. Xiao et al. [21] conducted experiments on the Plantvillage dataset, utilizing different feature encoders within the Prototypical Network, Matching Network, and Relational Network. They obtained average accuracies of 77.60%, 73.01%, and 73.13%, respectively, for 5way-1shot. The results referred to assessing the feasibility of FSL in crop leaf disease recognition. Li et al. [22] conducted a cross-domain FSL study by mixing pest data and achieved high performance in crop leaf disease recognition. Lin et al. [18] proposed a network based on combining multi-scale features and channel attention to enrich feature representation. The method achieved high performance on 5way-1shot and 5way-5shot on the Plantvillage dataset. In summary, these studies have demonstrated the potential viability of FSL for crop leaf disease recognition, but challenges remain in crop leaf disease recognition with the aforementioned FSL-based methods [23]. Currently, three specific issues persist in the research on FSL-based recognition of foliar disease in crops. First, there is a significant distribution shift between the field and laboratory samples. The large amount of visual interference and irrelevant information in the field images increases the complexity of the intrinsic dimension of the feature space. However, the feature extraction networks of the above FSL methods generally have the ability to adaptively filter and process the distribution of the input samples, and it is difficult to maintain good generalization in the complex field feature space. On the other hand, crop leaf disease recognition belongs to fine-grained image recognition tasks. Different disease categories of the same crop exhibit similar features, which requires the model to have stronger feature spatial representation capability. However, current studies have paid less attention to how to obtain a feature extractor with stronger generalization from the perspective of training strategy. Nonetheless, when the sample size is extremely limited, such as with just one sample, the classification performance of a single FSL model tends to be subpar. Therefore, improving classification reliability under very few samples is a difficult challenge for FSL.

To address the above issues, this study proposes a HMFN-FSL framework that incorporates CBAM [24] attention mechanism. Firstly, considering the characteristics of crop leaf spots, this study introduces the CBAM attention module into the feature encoder to focus on the key regions; secondly, this study optimizes a pre-training strategy for meta-learning to improve the generalization of feature representations. Finally, the classification performance is improved by constructing a Heterogeneous Metrics Fusion Network for FSL (HMFN-FSL), which fuses multiple FSL models with appropriate weighting. The aim of the study objective is to improve the ability of the FSL method to recognize crop leaf diseases, thus providing an effective method for recognizing crop leaf diseases in the field with small samples. The contributions of this study can be summarized as follows:

(1) A CBAM attention module was embedded in the feature network for few-shot learning to focus on important lesion feature regions to improve the generalization ability of the network.
(2) Optimization of the pre-training strategy of the feature extraction network for metric learning to improve the generalization ability of the feature encoder.
(3) A Heterogeneous Metrics Fusion Network (HMFN-FSL) was constructed to improve the prediction performance and reliability.
(4) Extensive experiments were conducted on crop leaf disease datasets in laboratory scenes and field scenes to validate the superiority of the model and to provide a feasible solution for the recognition of crop leaf diseases for few-shot learning in the field.

2. Materials and Methods

2.1. Dataset

Three public datasets were used in this study: Mini-ImageNet [25], Plantvillage [26], and Field-PV [27]. Plantvillage is one of the most frequently used datasets in crop leaf disease recognition studies [27–31], containing 50,403 images covering 38 categories for 14 crops. In addition, a field scenes disease dataset named Field-PV was selected for this study. This dataset has the same distribution of disease species as the Plantvillage dataset, as shown in Table 1. The obvious difference between the Field-PV dataset and Plantvillage is that its background is more complex and it is closer to natural field scenes, so that it can be used to simulate the recognition of crop leaf diseases in field scenes. Specifically, we selected four representative diseases (as shown in Figure 1), namely corn leaf blight, grape black measles, potato early blight, and strawberry leaf scorch. From the two datasets for comparison, we can see that the Field dataset's background is more complex than that of the Plantvillage dataset. The disease images contain background disturbances such as healthy leaves, sunlight, and soil.

Table 1. The 14 species and 38 categories in PV and Field-PV.

Species	Class Numbers	Class Name	Number of PV	Number of Field-PV
Apple	4	Apple scab, black rot, cedar, healthy	3174	72
Blueberry	1	Healthy	1502	12
cherry	2	Healthy, powdery mildew	1905	20
corn	4	Gray leaf spot, common gray leaf spot, common	3852	82
Grape	4	Black rot, black measles, healthy, leaf blight	3862	52
Orange	1	Haunglongbing	5507	33
Peach	2	Bacterial spot, healthy	2657	31
Pepper	2	Bacterial spot, healthy	2473	21
Potato	3	Early blight, healthy, late blight	2152	36
Raspberry	1	Healthy	371	8
Soybean	1	Healthy	5089	23
Squash	1	Powdery mildew	1835	25
Strawberry	2	Healthy, leaf scorch	1565	80
Tomato	10	Bacterial spot, early blight, Healthy, late blight, leaf mold, septoria leaf spot, spider	18,159	169

2.2. Problem Formulation

2.2.1. Support Set and Query Set

In few-shot learning (FSL), the dataset is divided into base classes C_{base} and novel classes C_{novel}. The base class data are used for meta-training, while the novel class data are used for meta-testing. The training data do not intersect with the categories of the test data, $C_{base} \cap C_{novel} = \varnothing$, which means that the categories of the test data are not visible to the training process. Since the novel class only has a small number of samples, the data are organized in the form of tasks. Specifically, the tasks that are sampled at a time are called episodes, and each episode consists of an S (support set) and a Q (query set). It can be expressed as:

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \tag{1}$$

$$Q = \{(x_1, y_1), \ldots, (x_n, y_n)\} \tag{2}$$

where (x_s, y_s) is an image label pair. In the formula, m is the number of samples in the support set and n is the number of samples in the query set. The model is trained using the labels from Q to compute the loss and thus perform supervised learning.

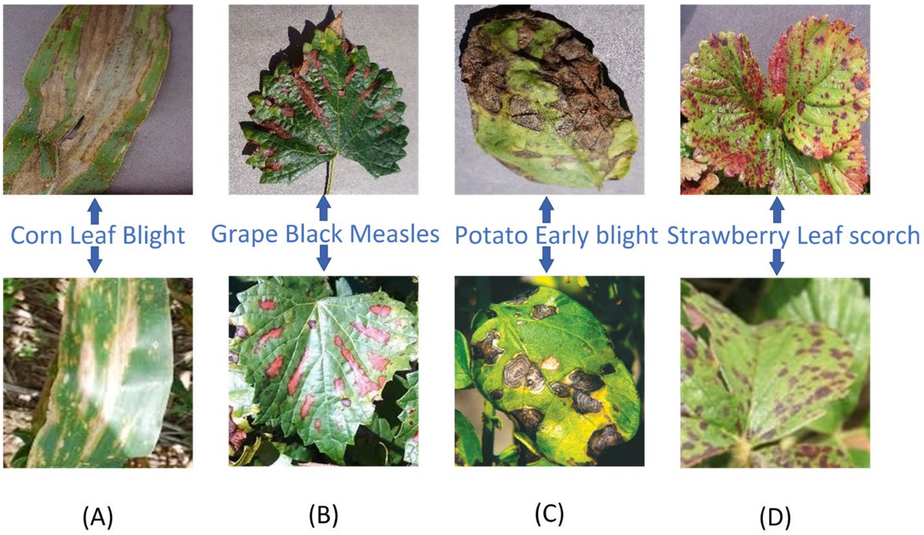

Figure 1. Comparison of images in different datasets. (**A**) Corn leaf blight. (**B**) Grape black measles. (**C**) Potato early blight. (**D**) Strawberry leaf scorch.

2.2.2. N-Way K-Shot

N-way K-shot means that there are N categories in the support set and only K support samples in each category. In simple terms, the parameter K indicates exactly how few samples there are in each category in a recognition task with N categories.

2.3. Preparatory

2.3.1. Metric Learning

For data-driven deep learning, the N-way K-shot task is a very difficult scenario. Since only K samples are available for learning in the task, it is difficult to achieve good results for either transfer learning or data augmentation in this scenario. However, with the continuous emergence of FSL based on meta-learning, this problem has been solved, which also makes meta-learning the mainstream method to solve the FSL problem at once. Among these methods, metric learning is currently one of the more effective methods.

The right side of Figure 2 shows the overall metric learning framework. The core idea of metric learning is to compute the distance matrix between the support set and the query set by the metric module and achieve classification by using the distance similarity between the samples. The model proposed in this study uses three robust metric networks: the Prototypical Network, the Matching Network, and the DeepEMD.

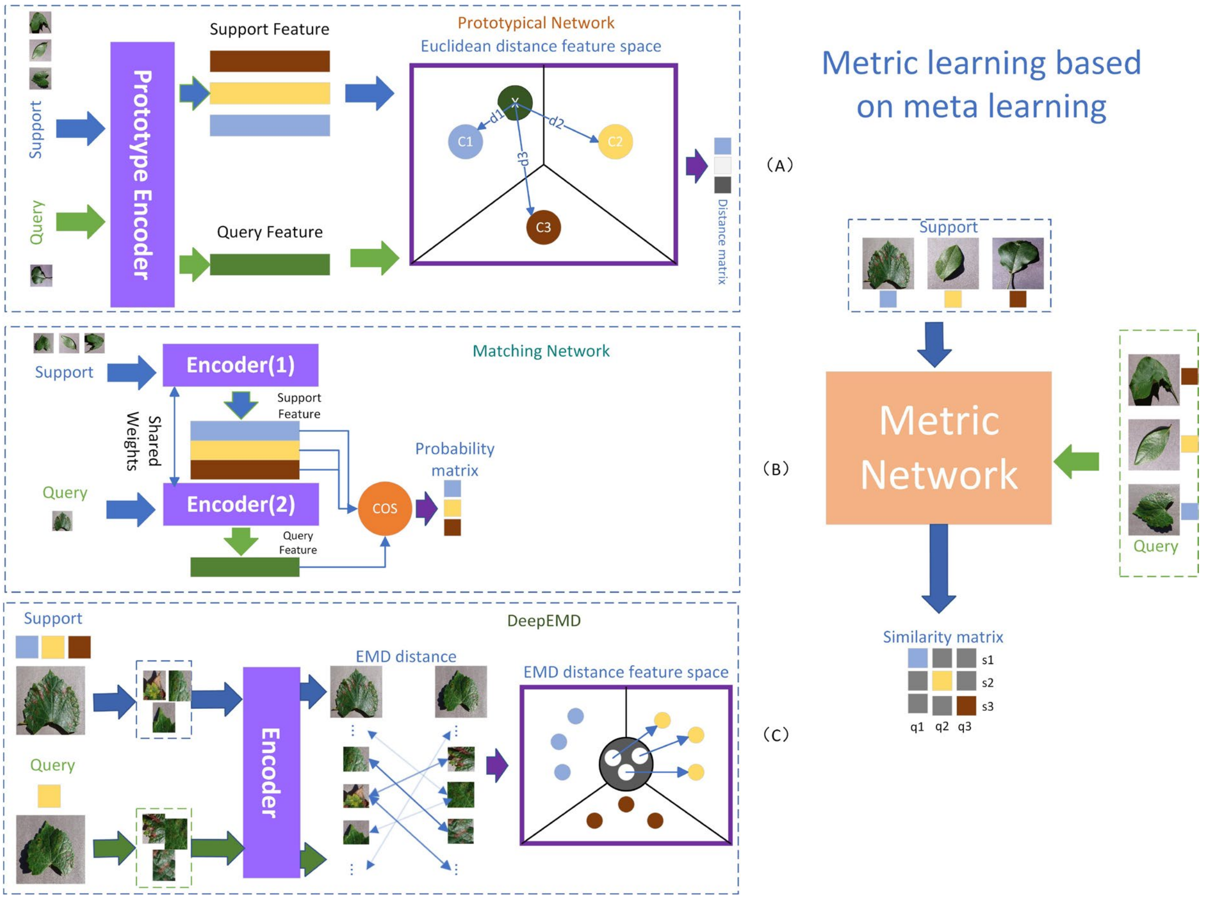

Figure 2. The framework of metric learning. (**A**) Prototypical Network. (**B**) Matching Network. (**C**) DeepEMD.

- Prototypical Network

Figure 2A shows the Prototypical Network proposed by Snell et al. [32]. It projects the samples through an encoder into the high-dimensional space. Subsequently, it computes the mean centers of the support set in this space, denoted as C_1, C_2, C_3. It computes the mean centers of the support set in this space, denoted as X, into the same feature space. It then calculates the Euclidean distance between the mean centers, named the Average Prototype (AP), and X. Finally, the query is classified based on the magnitude of the distance between it and the three Average Prototypes (APs), denoted as d_1, d_2, d_3. Thus, the classification problem is transformed into a spatial nearest-neighbor problem.

- Matching Network

As shown in Figure 2B, the Matching Network proposed by Oriol et al. [33] is the first meta-learning model based on metric learning. First, it extracts features from the support set and the query set through the Long Short-Term Memory (LSTM) network [34]; this is achieved by exploiting its memory capability in sequence modeling, so that the model can comprehensively consider the features of both support samples and query samples. Finally, the cosine distance between the features of the support set and the features of the query set is computed, and the probability distribution of the query set belonging to each category is obtained after the SoftMax operation to achieve the classification. The Matching Network obtains the mapping of the input space to the metric space through set-to-set learning [35] and provides an FSL method based on metric learning and the external memory.

- DeepEMD

Shown in Figure 2C is the DeepEMD proposed by Chi et al. [36]. EMD (Earth Mover's Distance) is an image similarity measure. Specifically, features are extracted from the support set and query set to obtain feature vectors of length m and k, respectively.

$$A_{ij} = 1 - \frac{{S_i}^T D_j}{||S_i|| \|D_j\|} \tag{3}$$

$$\text{Minimize :} \sum_{i=1}^{m} \sum_{j=1}^{k} X_{ij} A_{ij} \tag{4}$$

$$\mathit{Subject\ to:}\ \sum_{j}^{k} X_{ij} = S_i \tag{5}$$

$$\sum_{i}^{m} X_{ij} = D_j \tag{6}$$

where the product of the number of feature maps and the feature map size of the support set image is k, while the query set is m. Then, m and k can be considered as the number of places of suppliers and demanders. The unit cost of transport between vectors A_{ij} (as in Equation (3)) is expressed by the cosine distance. Thus, a linear programming problem with EMD with constraints of Equations (5) and (6) is constructed for the inter-image between the support set and query set. The optimal matching cost between the images (as in Equation (4)) can be obtained by requiring the transport volume X_{ij} from each supplier i to each demander j. EMD constructs a spatial nearest-neighbor problem under the local feature space of an image by computing the minimum matching cost in Equation (4) and using it as the distance in the prototypical network.

2.3.2. CBAM (Convolutional Block Attention Module)

CBAM (Convolutional Block Attention Module) is a lightweight attention module consisting of a Channel Attention Module (CAM) and a Spatial Attention Module (SAM). CBAM can perform both spatial and channel attention operations to make the feature encoder pay more attention to targets, which is the area of the crop leaf disease spot.

- Channel Attention Module

The Channel Attention Module (CAM) focuses on semantic concepts in the image that are more relevant to the current classification task by modeling the correlations within the feature channels. The structure of the Channel Attention Module, as shown in Figure 3A, is as follows: first, the input feature maps are subjected to both maximum pooling and average pooling in order to capture the global and salient features of the feature maps in the channel dimension, respectively. The pooling results are fed into a convolutional network with two layers of shared weights. In this process, down-sampling and up-sampling operations are performed on the feature maps in order to obtain more complex feature information. Afterwards, the output is then summed element-by-element and non-linearly transformed using Sigmoid to obtain a weight map with just one channel representing the allocation of attention weights to different channels. This attentional feature map highlights the features in the input feature map that are more relevant to the current task in the channel dimension. Eventually, the result of the inner product operation of the attention feature map with the original feature map is used as the output to achieve the effect of enhancing valid features to suppress invalid channel features. This structure allows the model to automatically learn salient information about the target at the channel level.

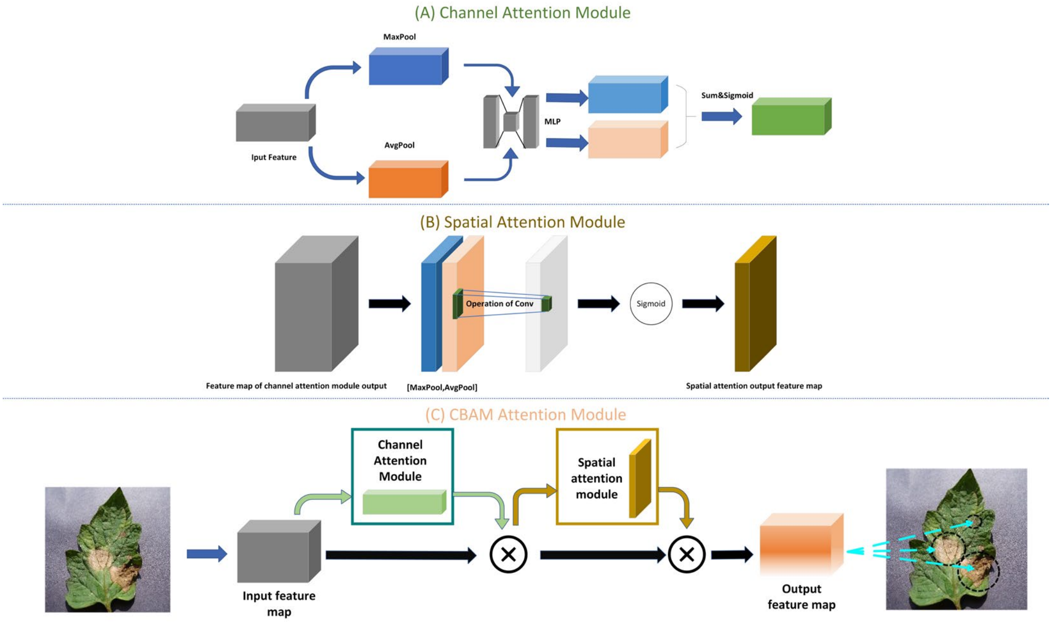

Figure 3. CBAM structure diagram. (**A**) The structure of the Channel Attention Module. (**B**) The structure of Spatial Attention Module. (**C**) The computational flow of CBAM.

- Spatial Attention Module

The Spatial Attention Module (SAM) focuses on the spatial distribution of valid information in an image by modeling the relationships within the feature space. Spatial attention enables the model to focus on the effective region in the image by learning the correlation inherent in the feature space, thus improving the model's ability to recognize and process local details, and its structure is shown in Figure 3B. First, average pooling and maximum pooling are performed on the channel dimension, and then their feature maps are concatenated to fuse different types of feature information. The concatenated feature maps are convolved through a layer to generate a spatial weighting allocation feature map, and the prominent parts of the feature map are assigned weights using the sigmoid function. Finally, the spatial feature maps are subjected to an inner product operation with the original feature maps to achieve weighting of the input feature maps for spatial attention feature extraction. This structure allows the model to focus on the effective regions in the image by learning the intrinsic spatial correlation of the input feature maps.

- CBAM

As shown in Figure 3C, the CBAM module is a feedforward structure consisting of a Channel Attention Module (CAM) and a Spatial Attention Module (SAM) connected in series. Firstly, the original feature maps are used as the input of the CAM module, and the feature maps in the channel dimension are learned first. Then, the output of the CAM is used as the input of the SAM module to learn the feature map in the spatial dimension. In this way, the feature map obtains appropriate customized features in both the channel and spatial dimensions. Since the input and output feature maps of the CBAM module have the same shape, it can be seamlessly embedded into the tail of the residual block of the ResNet network [35]. In this study, we refer to the ResNet network with the combined CBAM module in the article as CBAM-ResNet.

2.3.3. Stacking Framework

Stacking is a widely used multi-model fusion technique [37–39]. The framework usually consists of two parts: a base learner and a meta-learner. As shown in Figure 4, multiple base learners output their respective predictions. These predictions are pooled as training inputs for the meta-learner. Finally, the parameters in the base learner and the meta-learner are updated using the real labels of the original dataset as supervision. The base learners used in this study are the Prototypical Network, the Matching Network, and the DeepEMD. The significant advantage of stacking is that it can utilize the strengths of different base learners to compensate for their respective weaknesses, thus improving the accuracy of the final prediction. This type of integration can effectively enhance the stability and the generalization ability of the model.

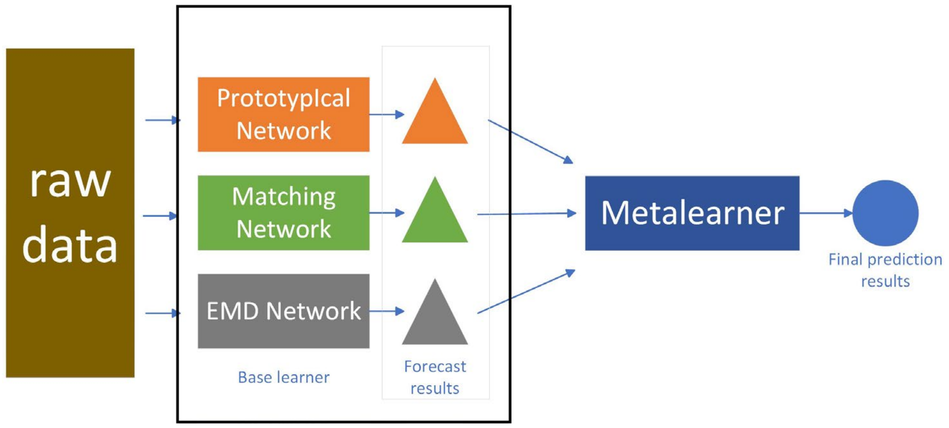

Figure 4. The flow chart of stacking. The raw data go through the three base models and output the predictions, which are fed into the meta-learner to obtain the final outputs.

2.4. The Architecture of HMFN-FSL

In this subsection, we will describe the meta-learning framework of HMFN-FSL, how the loss function takes into account the parameters in the three networks, and the pseudocode of HMFN-FSL.

2.4.1. The Meta-Learning Framework of HMFN-FSL

For the Prototypical Network, Matching Network, and DeepEMD, in order to realize the shift from traditional classification learning to metric learning, each base learner replaces the linear classifier used in the pre-training phase (e.g., Figure 5A) with the distance metric module of the corresponding model. In other words, the classification result is completely determined by the distance between the support set and the query set, and not by the linear discrimination of the classifier. Under the N-way K-shot task setting, for category c, this study randomly samples K samples of this category from the support set S and inputs them into the feature extractor $F(\theta_n)$ for feature extraction. The feature mean of these K samples is then computed as the prototype vector T_C of the category, which is used to represent the category c.

$$T_C = \frac{1}{|S_c|}\sum_{x_s \in S_C} F_\theta(x_s) \tag{7}$$

where S_C represents the number of samples in the support set in category c. Meanwhile, the samples in the query set Q undergo feature extraction by $F(\theta_n)$ to obtain q high-dimensional feature vectors x_q. Then, the probability that q belongs to category c can be expressed as:

$$p(y = c|x_q) = \frac{\exp(-d(F_\theta(x_q), T_C)}{\sum_{C'} -d(F_\theta(x_q), T_{C'})} \tag{8}$$

where $T_{C'}$ represents the center vectors of all the categories; $d(F, T)$ represents the distance between the vector of query set F and center vectors of the T category. The cross-entropy loss function is computed in training by back-propagating it to the encoder $F(\theta_n)$ by adjusting θ_n to achieve the purpose that the different categories are distant from each other and the same categories are close to each other. The loss can be expressed as:

$$Loss = d(F_\theta(x_q), T_{C'}) + log\sum_{C'} \exp(-d(F_\theta(x_q), T_{C'}) \tag{9}$$

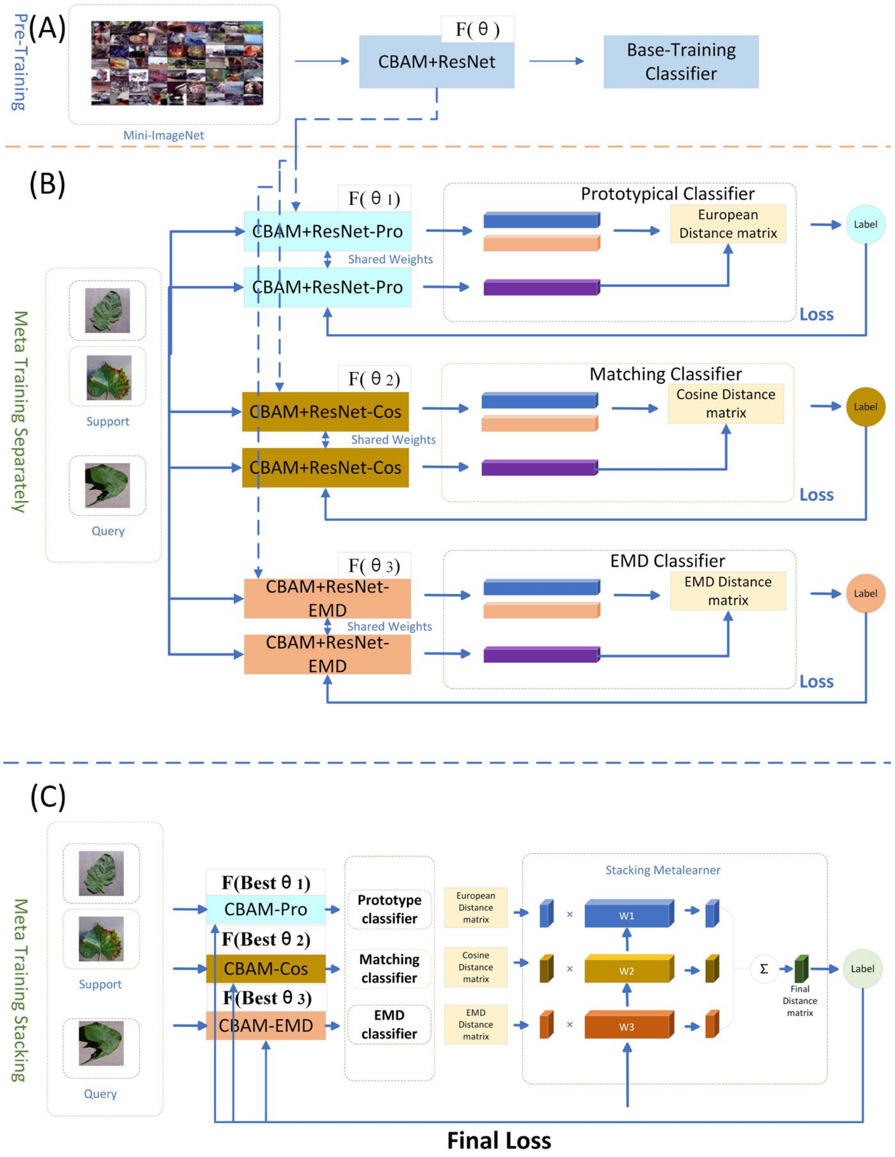

Figure 5. The network architecture of HMFN-FSL. (**A**) Pre-training phase: pre-training the encoder through classification tasks to obtain feature representations with generalization ability. (**B**) Meta-training separately phase: train multiple base learners separately, including Prototype Networks, Matching Networks, and DeepEMD. (**C**) Meta-training stacking phase: input the distance matrix output by each base learner to the weighted fusion meta-learner and jointly fine-tune the parameters of the base learners by back-propagation.

In the meta-training separately phase, this study performs the above training process for each model (e.g., the Prototypical Network) individually until its convergence, in order to obtain the individual optimal metric learning capability to provide a reliable base learner for subsequent meta-learning.

As shown in Figure 5C, during the meta-training stacking phase, the encoder parameters of the three base learners are initialized to the optimal values of θ_1, θ_2, and θ_3 obtained in their respective meta-training separately phases. At the same time, the three base learners (Prototype Network, Matching Network, and DeepEMD) output the distance matrices D_A, D_B, and D_C, respectively. However, these three distance matrices are not on a uniform scale due to the adoption of different distance metric functions. In order to realize the linear fusion of the distance matrices, this study cleverly adopts SoftMax to normalize them (see Equation (8)) and maps the three distance matrices to a unified metric space in a probabilistic way.

Subsequently, the three normalized distance matrices are weighted and fused using the fusion network, to obtain the final distance matrix D_M:

$$D_M = W_1 \cdot D_A + W_2 \cdot D_B + W_3 \cdot D_C \tag{10}$$

where W_1, W_2 , and W_3 are the weighting parameters assigned by the meta-learning, respectively. Based on D_M, the cross-entropy loss function Loss-D_M is computed and backpropagated to the four networks to update the parameters of encoder in the base learners as well as the parameters of the meta-learners. The final distance matrix D_M is affected by both the parameters θ_1, θ_2, θ_3 of the three base learners as well as the meta-learner parameter θ_4. In this study, we denote all the parameters as $\varnothing(\theta_1, \theta_2, \theta_3, \theta_4)$; then, Loss-$D_M$ can be expressed as:

$$Loss\ D_M = d\left(F_\varnothing\left(x_q\right), T_{C'}\right) + log \sum\nolimits_{C'} \exp\left(-d\left(F_\varnothing\left(x_q\right), T_{C'}\right)\right. \tag{11}$$

Meanwhile, the probability of belonging to the category c for q can be expressed as:

$$p\left(y = c \middle| x_q\right) = \frac{\exp\left(-d\left(F_\varnothing\left(x_q\right), T_C\right)\right.}{\sum_{C'} -d\left(F_\varnothing\left(x_q\right), T_{C'}\right)} \tag{12}$$

This can realize the recalibration and fusion of the distance metric results of each base learner, so that the model can synthesize the advantages of different distance metrics and achieve stronger few-shot learning performance.

2.4.2. Feasibility of Loss-$\mathrm{D_M}$

Meanwhile, in order to verify whether the prediction results of all base learners can be considered simultaneously in the back-propagation process of the loss function Loss-D_M, the following derivation is performed in this study.

When Loss-D_M is used to update the parameters of the nth model, it uses the combined output matrix D_M to calculate the gradient, which is the sum of the output matrices of the three models. Theoretically, the gradient of the nth model will be affected by the contribution of all models in the integrated learning to the final output and will be adjusted accordingly. In this study, we denote the partial derivative of the loss function with respect to the parameter θ_n of the nth model as $\frac{d\,L}{dF_{\theta_n}}$. And we use the chain rule for partial derivatives to express this derivative as:

$$\frac{d\,L}{d\,F_{\theta_n}} = \frac{d\,L}{d\,D_M} \times \frac{d\,D_M}{d\,F_{\theta_n}} \tag{13}$$

where $\frac{d\,L}{d\,D_M}$ is the partial derivative of the loss function with respect to the combined output matrix D_M and $\frac{d\,D_M}{dF_{\theta_n}}$ is the partial derivative of the combined output matrix D_M with respect to the parameter θ_n of the nth model. Therefore, the partial derivatives of D_M with respect to the parameter θ_A of model A can be computed as:

$$\frac{d\,D_M}{d\,F_{\theta_A}} = \frac{d\,(W_1 \cdot D_A + W_1 \cdot D_B + W_3 \cdot D_C)}{d\,F_{\theta_A}} = W_1 \cdot \frac{d\,D_A}{d\,F_{\theta_A}} \tag{14}$$

The partial derivatives of the components in D_B and D_C and the stacking meta-models in D_M with respect to θ_A should be zero since they are obtained by linearly stacking D_M so they are a constant with respect to θ_A, which is obtained by inserting the above equation into Equation (10):

$$\frac{d\,L}{d\,F_{\theta_A}} = W_1 \cdot \frac{d\,L}{d\,D_M} \times \frac{d\,D_A}{d\,F_{\theta_A}} \tag{15}$$

This indicates that the partial derivatives of the loss function with respect to the $\boldsymbol{\theta_A}$ component of the parameters in model A are composed of three parts. The updating of the parameters in the A network via Loss-D_M, which references both the performance of $\frac{d\ D_A}{d\ F_{\theta_A}}$ in its own predictions and the presence of $W_1 \cdot \frac{d\ L}{d\ D_M}$ as an externally variable constant, suggests that the A model is updated with reference to the synergistic contributions of the other three models to the final output of the base model A as well. Similarly, it can be inferred that the parameter updates of other models also depend on their own effects as well as those of other models. In this way, a synergistic optimization is achieved between the individual base learners in the meta-learning framework.

2.4.3. Algorithm of HMFN-FSL

In order to increase the reproducibility of our method, the pseudocode of the algorithm of this study is shown in Algorithm 1.

Algorithm 1 The algorithm of HMFN-FSL

```
Input: dataloader, n_way, n_shot, n_query, task_per_batch
Output: avg_acc, avg_loss
for i in epoch:
  train:
  for j in batch:
    task = task(dataloader, n_way,n_shot, n_query, task_per_batch)
    $X_1 0 \ldots X_1 n = f_{\theta 1}$(task.x_shot)
    $X_1$ = mean($X_1 0 \ldots X_1 n$)
    $y_1 = f_{\theta 1}$(task.x_query)
    $X_2 0 \ldots X_2 n = f_{\theta 2}$(task.x_shot)
    $X_2$ = mean($X_2 0 \ldots X_2 n$)
    $y_2 = f_{\theta 2}$(task.x_query)
    $X_3 0 \ldots X_3 n = f_{\theta 3}$(task.x_shot)
    $X_3$ = mean($X_3 0 \ldots X_3 n$)
    $y_3 = f_{\theta 3}$(task.x_query)
    Logits_Proto = classifer(Proto_distance(x1, y3))
    Logits_Matching = classifer(Matching_distance(x2, y3))
    Logits_EMD = classifer(EMD_distance(x3, y3))
    Total_Logits = HMFN-FSL$_{\theta 4}$(Logits_Proto, Logits_Matching, Logits_EMD)
    Loss = cross_entropy(Total_Logits, task.label)
    acc = compute (Total_Logits, task.label)
    Loss.backward()
  end for
  Validation: val
  Compute: avg_acc, avg_loss
end for
Return: avg_acc, avg_loss
```

3. Results

In this study, comparative experiments and ablation experiments are used to illustrate the effectiveness of the methodology proposed in this study as well as the effect of changes in various variables on the accuracy of the model. Specific experiments and results are described and analyzed in detail below.

3.1. Data Setting

As shown in Figure 6, in order to satisfy the requirement of setting the training set and test set categories as mutually exclusive in Section 2.2, the disease categories of the Plantvillage dataset are sorted alphabetically in this study, and the training set, validation set, and test set are partitioned. Specifically, the odd-numbered categories of the sorted dataset (a total of 19 disease categories) were used as the training set, the odd-numbered categories of the remaining categories (10 disease categories) were used as the validation set, and the remaining 9 categories were used as the test set. Meanwhile, in order to fully evaluate the recognition performance of the model on the Field-PV dataset, three different splits were applied to these data. (1) Field-PV-all: all categories in Field-PV were included in the test set in order to compare the results with those in Plantvillage laboratory conditions. (2) Field-PV-split1: five representative diseases of different crops were randomly selected in Field-PV to form the test set to evaluate the performance of the model in recognizing different crop leaf diseases. (3) Field-PV-split2: five tomato diseases with higher complexity in Field-PV were selected as the test set to verify the model's recognition performance on fine-grained disease categories. With these three different test sets, this study is able to comprehensively evaluate the robustness and generalization ability of the model on Field-PV, which is a field scenes dataset.

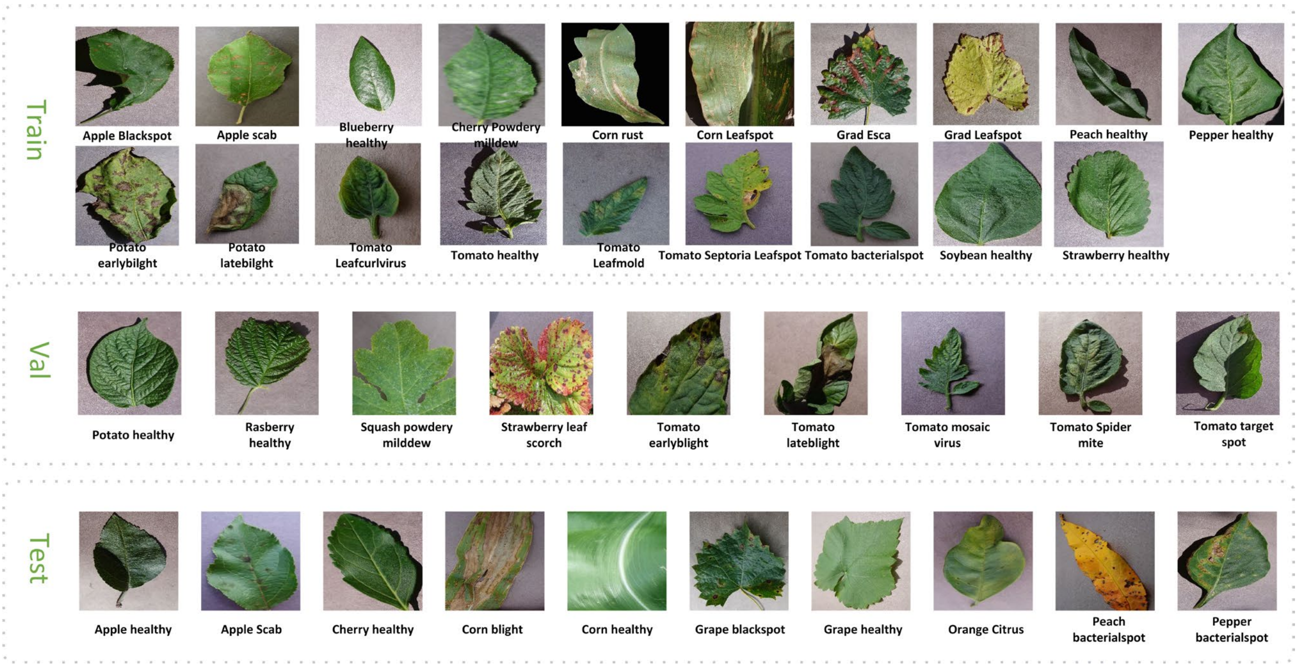

Figure 6. Map of Plantvillage dataset partitions. Train, Val, and Test are the disease categories and their pictures contained in the training, validation, and test sets, respectively.

3.2. Training Strategy and Hyperparameters

The algorithmic process in this study is roughly divided into three main phases: pre-training, meta-learning, and meta-testing phases. In order to comprehensively evaluate the impact of using different datasets in different phases on the performance of the model, six dataset configurations of S1–S6 are designed in this study (as shown in Figure 7). S1 uses Mini-ImageNet in both pre-training and meta-learning phases, and it performs meta-testing on the test set of Plantvillage. S2 uses Mini-ImageNet in the pre-training phase and the train sets and test sets of Plantvillage in the meta-learning and meta-testing phases, respectively. S3 uses the Plantvillage dataset in all the three phases. S4 uses Mini-ImageNet for pre-training, the train set of Plantvillage for meta-learning, and all Field-PV categories for meta-testing. S5 uses Mini-ImageNet for pre-training, Plantvillage for meta-learning, and the Field-PV-split1 for meta-testing. The data used for pre-training and meta-learning in S6 are the same as S4 and S5. The difference is that Field-PV-split2 is used for the meta-testing phase. By configuring these six strategies, this study can explore the best pre-training strategies and the migration effects of models in laboratory and field scenes. In addition, it can also explore the generalization ability of the model in different application scenes.

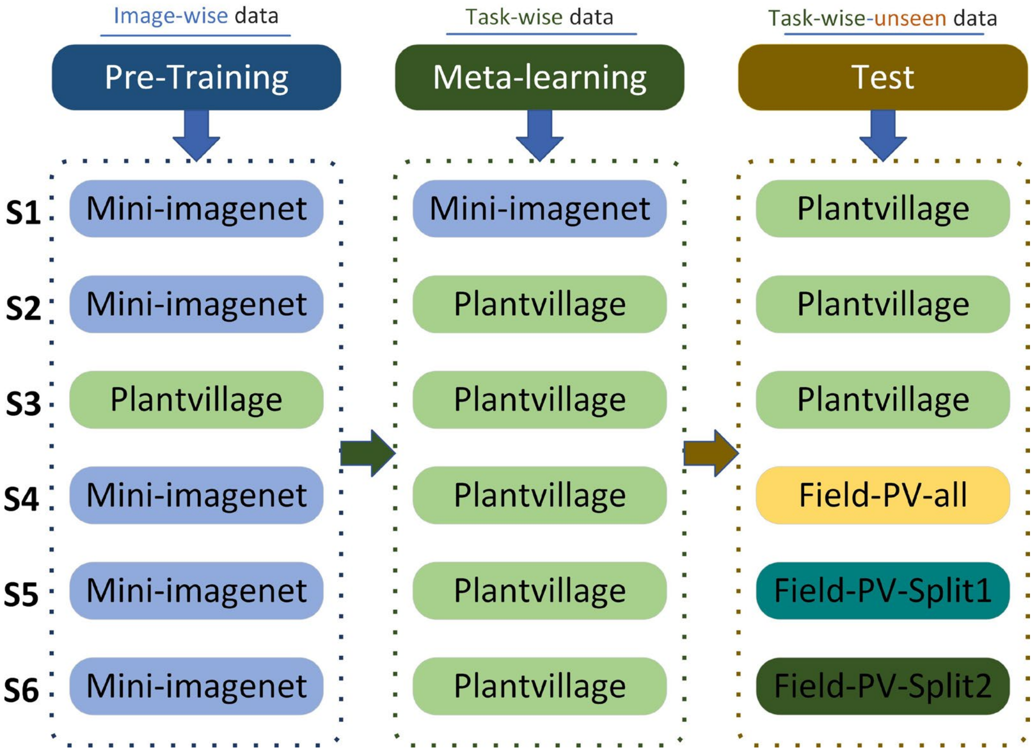

Figure 7. The data formats used in pre-training, meta-learning, and test. The left side represents the dataset used for pre-training, the center represents the dataset used for meta-training, and the right side is the dataset used for testing.

The hyperparameters for model training were then set as follows: in the pre-training phase, an SGD optimizer used in [40] was used with a momentum parameter of 0.9 and a weight decay factor of 5×10^{-4}. The initial learning rate was set to 1×10^{-2}, and the weights were decayed with the multiplicity of 0.2 every 20 rounds for a total of 100 training rounds. In the meta-training stacking, the base learners still used the SGD with an initial learning rate of 5×10^{-3}. In contrast, the meta-learner used the AdamW optimizer used in [36] with an initial learning rate of 1×10^{-3}. The learning rates of both the base learner and the meta-learner decayed with a multiplier of 0.5 every 30 rounds. In the meta-training stacking, the base learners still used the SGD with an initial learning rate of 5×10^{-3}, while the meta-learner used the optimizer named AdamW used in [41] with an initial learning rate of 1×10^{-3}, and the learning rates of both the base learner and the meta-learner decayed with a multiplier of 0.5 every 30 rounds. The average accuracy and the 95% MSE (mean square error) of 600 episodes used in [42] were measured in the testing phase.

3.3. CBAM Effectiveness

3.3.1. The Performance of Base Learners with CBAM

In order to verify the effectiveness of CBAM on the three underlying networks, we conducted a total of six sets of ablation experiments under the S3 data strategy on three base learners by controlling the presence or absence of CBAM in the backbone network. According to Table 2, upon integrating CBAM, the Prototypical Network, Matching Network, and DeepEMD achieved accuracies of 74.55%, 74.61%, and 78.22%, respectively. The accuracy of three networks increased by 2.34%, 2.10%, and 1.98%, respectively. The result confirms that the CBAM module can effectively improve the performance of disease recognition in basic metric learning models.

Table 2. Experimental results of CBAM effectiveness.

ID	Encoder	Method	5way-1shot (Acc (%))	5way-1shot (95% MSE)
E1	ResNet	Prototypical Net	72.21	0.12
E2	CBAM-ResNet	Prototypical Net	74.55	0.05
E3	ResNet	Matching Net	72.51	0.10
E4	CBAM-ResNet	Matching Net	74.61	0.12
E5	ResNet	DeepEMD	76.34	0.10
E6	CBAM-ResNet	DeepEMD	78.22	0.07

In order to increase the interpretability of CBAM-ResNet, this study uses the Grad-CAM visualization technique in the form of heat maps to show the degree of attention of the pre-trained network to different regions in the image recognition process. Figure 8 takes potato early blight as an example; Figure 8A shows the original image, and Figure 8B,C shows the heat maps of CBAM-ResNet and ResNet, respectively. The highlighted areas in red on the heat map indicate the areas of higher model attention. It can be seen that regions a and b in both CBAM-ResNet and ResNet are mainly focused on the root and lesion regions of potato leaves, which indicates that potato disease recognition mainly relies on the bases and lesion sites of crop leaves. Meanwhile, it can be noted that the a and b regions in Figure 8B exhibit higher brightness compared to the corresponding regions in Figure 8C. This suggests that, in comparison to ResNet, CBAM-ResNet pays more attention to the lesion areas of the original image, which verifies the effectiveness of the CBAM attention mechanism. Moreover, both networks attend to the background region; however, the c region of the CBAM-ResNet heat map is noticeably darker than that of ResNet. The highlighted region in the lower left corner demonstrates the same pattern. The experimental results of Algorithm 1 concur with this, showing that the robustness of the model improves and the influence of background noise reduces after integrating CBAM.

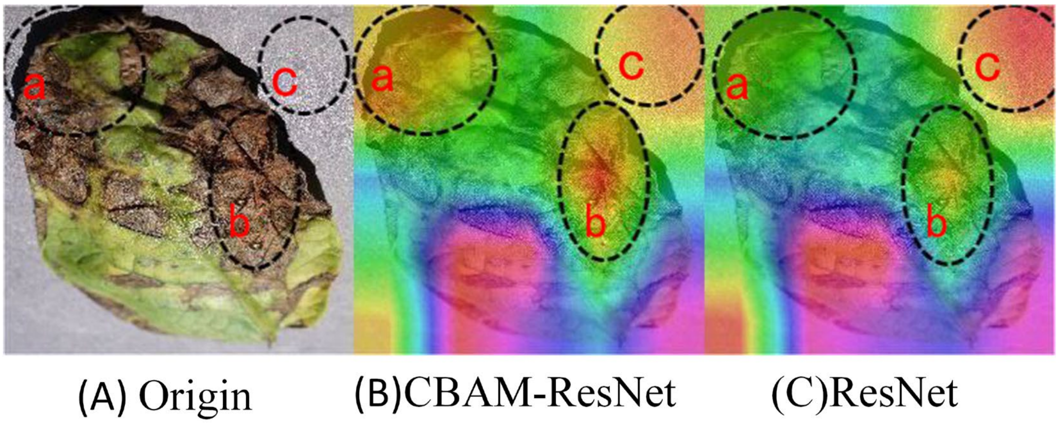

Figure 8. Heatmap of potato disease. (**A**) Original images of potato disease. (**B**) Heatmap of the feature map after adding CBAM. (**C**) Heatmap of the feature map before adding CBAM.

3.3.2. Base Learner Training Loss Curve

Loss function curves are shown in Figure 9. Figure 9A–C shows the loss function curves in the independent learning phase of the base learner, respectively. We can observe that although it is after the pre-training phase, the training loss of the model at the beginning of the meta-training phase is still large. This is because the feature space differentiation is still not distinct enough despite the pre-training. However, as the distance of different samples in the feature space becomes greater, the train losses of DeepEMD, Prototypical Network, and Matching Network gradually converge to 0.15, 1.30, and 0.9, respectively, and the value losses converge to 0.75, 1.40, and 1.0. Basically, the models start to converge within 60 rounds.

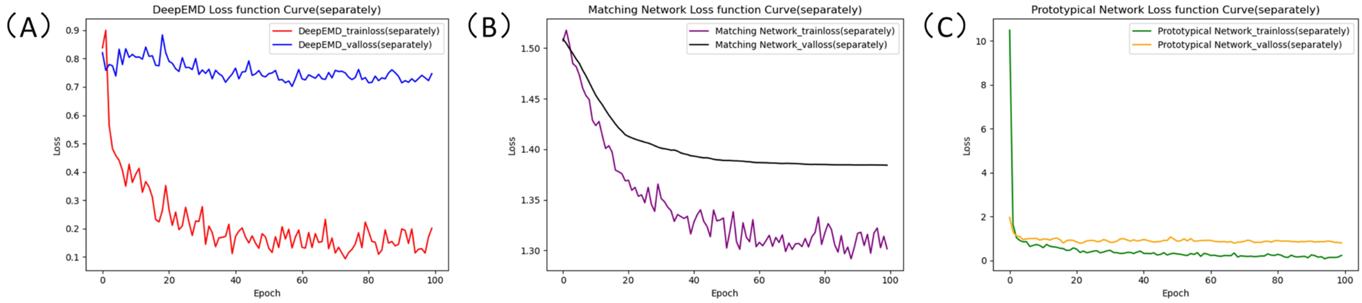

Figure 9. Base learner training loss curve variation. (**A**) DeepEMD separate training loss curve. (**B**) Matching Network separate training loss curve. (**C**) Prototypical Network separate training loss curve.

3.4. The Impact of Training Strategy

The broader application of the model feature representation can be improved by implementing transfer learning [43–45]. In this study, two training strategies are presented: one that uses exclusively Plantvillage pre-training and another that additionally incorporates Mini-ImageNet pre-training. Since the Mini-ImageNet dataset is larger and more diverse, using it for pre-training should theoretically result in more generalized feature representations. In order to verify the effectiveness of this training strategy in few-shot learning, this study conducted comparative experiments on Prototypical Networks, Matching Networks, and DeepEMD using the training strategy as the independent variable and testing accuracy as the dependent variable. The specific results are shown in Table 3. The addition of Mini-ImageNet pre-training has resulted in a notable improvement in the accuracy of the three models by 0.79%, 0.65%, and 5.55%, respectively. Especially for DeepEMD, which has higher requirements for feature expression, the accuracy improvement reaches 5.55%. These results corroborate that using Mini-ImageNet for pre-training can effectively improve the generalization capabilities of the feature extractor.

Table 3. The impact of pre-training strategies on performance.

ID	Training Stage	Method	5way-1shot (Acc (%))	5way-1shot (95% MSE)
F1	S2	Prototypical Net	75.34	0.11
F2	S3	Prototypical Net	74.55	0.05
F3	S2	Matching Net	75.26	0.10
F4	S3	Matching Net	74.61	0.12
F5	S2	DeepEMD	83.77	0.13
F6	S3	DeepEMD	78.22	0.07

3.5. The Performance of HMFN-FSL

3.5.1. Comparison with the Baseline Method

In this section, we select the S2 training strategy for comparison experiments between HMFN-FSL and the base learner with the addition of CBAM. As shown in Table 4, the experimental results show that the tested accuracy of the HMFN-FSL model reaches 91.20%, which is significantly better than the accuracy of the three base learners. Compared with the best base learner, DeepEMD, HMFN-FSL achieves an accuracy improvement of 7.43%. This advantage indicates that the HMFN-FSL fusion framework designed in this study can effectively utilize the characteristics of each base learner to improve the overall performance of FSL. In summary, this study effectively integrates multiple metric networks by constructing the HMFN-FSL model. Thus, a more powerful few-shot classifier is obtained to provide an effective FSL method for the few-shot crop leaf disease recognition task.

Table 4. HMFN-FSL performance analysis experimental results.

ID	Method	5way-1shot (Acc (%))	5way-1shot (95% MSE)
G1	Prototypical Net	75.34	0.11
G2	Matching Net	75.26	0.10
G3	DeepEMD	83.77	0.13
G4	HMFN-FSL	91.20	0.13

3.5.2. Ablation Analysis

In order to accurately quantify the contribution of each module to HMFN-FSL, we conducted ablation experiments on HMFN-FSL and evaluated the model on 5way-1shot, 5way-5shot tasks by controlling for the use of Mini-Imagenet and the presence or absence of CBAM. The specific experimental results are shown in Table 5, where pre-training with Mini-Imagenet resulted in 3.27% and 0.98% increases in model accuracy under the 5way-1shot and 5way-5shot tasks, respectively, and using CBAM resulted in 1.46% and 0.27% increases in model accuracy, respectively. The ablation experiments demonstrate that both pre-training with CBAM and Mini-Imagenet can improve the performance of the model.

Table 5. The ablation analysis of HMFN-FSL.

Pre-Train on Mini-Imagenet	CBAM	5way-1shot		5way-5shot	
		Acc (%)	95% MSE	Acc (%)	95% MSE
		86.37	0.13	96.06	0.05
√		89.64	0.13	97.04	0.05
	√	87.83	0.12	96.33	0.07
√	√	91.20	0.13	98.29	0.03

3.5.3. K-Fold Cross-Validation

In order to better verify the robustness and generalization of the model, we evaluated the model using K-fold cross-validation. Specifically, we homogenized the training data into (A, B, C, D, E) five folds and trained them separately under S2 training strategy conditions. We obtained a total of five sets of accuracy data. The average accuracy rate was obtained by averaging them and using this as the K-fold accuracy rate. The specific experimental results are shown in Table 6; the five folds achieved 91.20% and 98.17% accuracies on 5way-1shot and 5way-5shot, respectively, with polar deviations of 1.15% and 1.9%, respectively. The experimental results show that our model has strong stability and generalization ability.

Table 6. Performance analysis of HMFN-FSL K-fold cross-validation.

Fold ID	Method	5way-1shot (Acc (%))	5way-5shot (Acc (%))
Fold A	HMFN-FSL	91.17	98.29
Fold B	HMFN-FSL	90.64	97.07
Fold C	HMFN-FSL	91.15	98.09
Fold D	HMFN-FSL	91.79	98.97
Fold E	HMFN-FSL	91.23	98.36
AVG	HMFN-FSL	91.20	98.17

3.6. Comparison with Related Models

To verify the superiority of the proposed method in this study, comparative experiments were conducted with several other state-of-the-art methods, including Prototypical Network [32], Matching Network [33], DeepEMD [36], FEAT [46], and MAML [42]. The experiments for all groups except group A6 were conducted under the S1 training strategy, i.e., the PlantVillage dataset was used in all three phases. While group A6 used Plantvillage in the pre-training phase with Mini-Imagenet, the meta-training and meta-testing phases used the S1 training strategy. The final experimental results are shown in Table 7. The results show that the algorithm proposed in this study has a significant advantage in both 5way-1shot and 5way-5shot conditions and the accuracy rates reach 91.20% and 98.29% in 5way-1shot and 5way-5shot conditions, respectively. Compared with other models, the best-performing DeepEMD improved by 14.86% and 3.96% in 5way-1shot and 5way-1shot, respectively. In the meta-testing stage, each base learner gives a separate score for belonging to a particular disease, and that disease type with the highest score after multiplying the scores of the base learners by their respective weights is the final output of the group of base learners. Thus, it can be shown theoretically that the combined decision result of multiple base learners improves the stability of the classifier compared to a single meta-learner. Figure 10 is a comparison between other methods and HMFN-FSL. Whether on the 5way-1 shot task or the 5way-5 shot task, the accuracy of HMFN-FSL is higher than that of other base learners. The result shows that the integration of multiple learners has a higher accuracy rate than related methods using a single learner. The above results indicate that the algorithm in this study achieves state-of-the-art among the algorithms related to few-shot crop leaf disease recognition.

Table 7. Different performances of the models.

ID	Method	5way-1shot		5way-5shot	
		Acc (%)	95% MSE	Acc (%)	95% MSE
A1	Prototypical Net	72.21	0.12	91.26	0.13
A2	Matching Net	72.51	0.10	89.47	0.14
A3	DeepEMD	76.34	0.10	94.33	0.07
A4	FEAT	75.25	0.03	92.01	0.03
A5	MAML	63.25	0.16	82.17	0.07
A6	HMFN-FSL	91.20	0.13	98.29	0.03

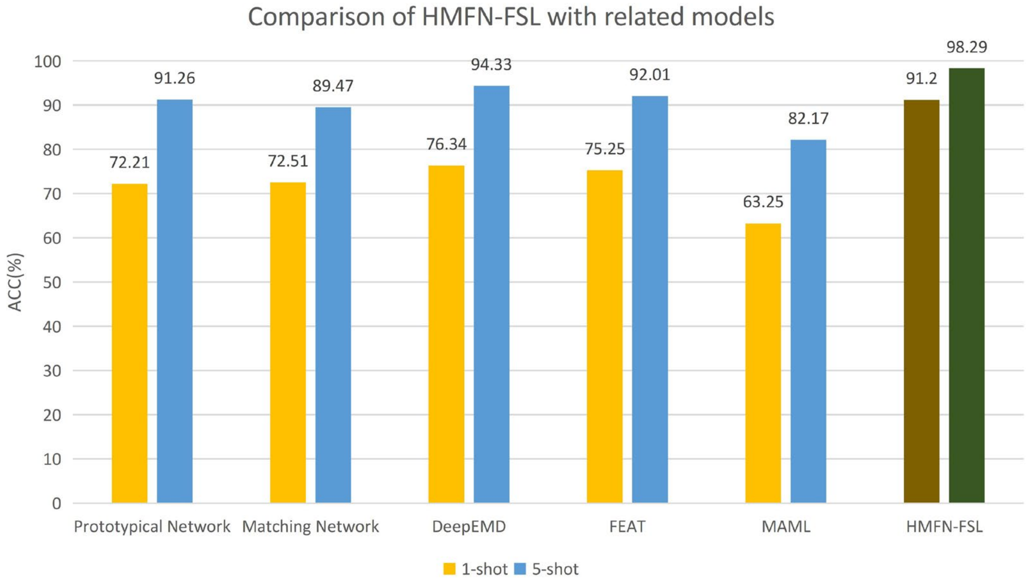

Figure 10. Comparison of HMFN-FSL with related models. Yellow columns and blue columns represent the accuracy of the comparison models 5way-1shot and 5way-5shot, respectively. In the last two bars, the brown columns and green columns represent the accuracy of 5way-1shot and 5way-5shot of HMFN-FSL.

3.7. Cross-Domain and Field Scenes

Cross-domain generalization capability is an important metric for evaluating models. In this subsection, the cross-domain generalization capability of the proposed method is explored from two aspects: (1) the migration from the non-crop leaf disease domain

(Mini-ImageNet) to the crop leaf disease domain (Plantvillage); (2) the migration from the laboratory scenes (Plantvillage) to the field scenes (Field-PV). Meanwhile, in this study, fine-grained recognition [47] experiments were conducted on five diseases of tomatoes (Field-PV-Split1) to further validate the cross-domain and fine-grained classification ability of the model. For this purpose, a total of five experiments were conducted in this study, as shown in Table 8.

Table 8. Cross-domain experimental results of the model.

ID	Training Stage	5way-1shot		5way-5shot	
		Acc (%)	95% MSE	Acc (%)	95% MSE
H1	S1	79.52	0.14	92.30	0.12
H2	S2	91.20	0.13	98.29	0.03
H3	S4	73.60	0.11	85.86	0.11
H4	S5	76.80	0.11	88.49	0.10
H5	S6	44.38	0.22	58.13	0.08

As shown in Figure 11, the experimental results show that in the meta-training stage, the 1-shot and 5-shot accuracies of the H1 group, which did not use any Plantvillage data, decreased by 11.68% and 5.98%, respectively, compared with those of the H2 group, which used Plantvillage data in the meta-training. This confirms that the size of the inter-domain differences directly affects the performance of the model and that the differences between the source and target domains should be as small as possible in practical applications. Notably, the performance of HMFN-FSL untrained by Plantvillage exceeds the performance of all the algorithms in Table 5. This is further evidence of the superiority of HMFN-FSL. In addition, the complexity of the scenes is also a key influencing factor. The H3 group is tested on the complex field scenes data and compared with the H2 group in the laboratory domain; the accuracies of 5way-1shot and 5way-5shot are 73.80% and 85.86%, respectively. There is an obvious performance decay, indicating that the complex scenes increase the recognition difficulty and place higher demands on the model's generalization ability. However, this result is still on par with the performance of all of the algorithms in Table 7 in the laboratory scenes. Even in the case of very limited training samples, this method can still get close to or even exceed the effect of lab algorithms under field sample conditions. This provides a strong guarantee for the intelligent diagnosis of crop leaf diseases from the laboratory to the field.

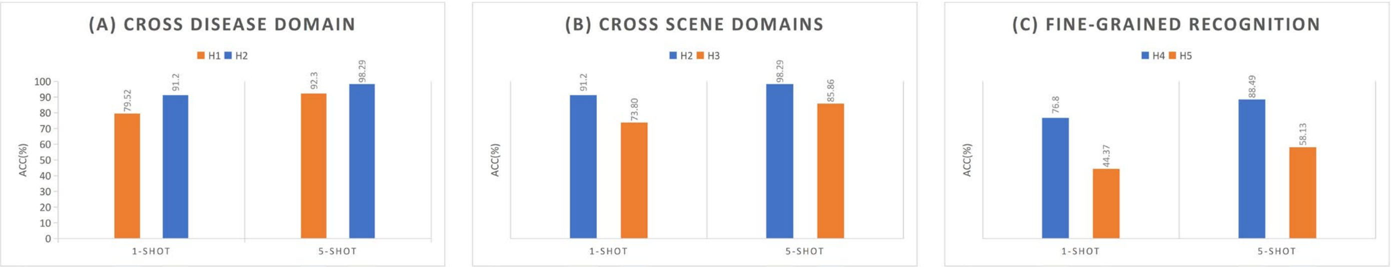

Figure 11. The performance of cross-domain and fine-grained recognition. (**A**) The performance of cross-disease domain. (**B**) The performance of cross-scenario domain. (**C**) The performance of fine-grained recognition.

Unfortunately, as shown in the confusion matrix in Figure 12, compared with coarse-grained recognition, fine-grained recognition only achieves 58.13% accuracy on 5way-5shot, and the number of errors increases from 345 to 1249, with a significant degradation in performance. This indicates that fine-grained crop leaf disease recognition in the field is still a challenge.

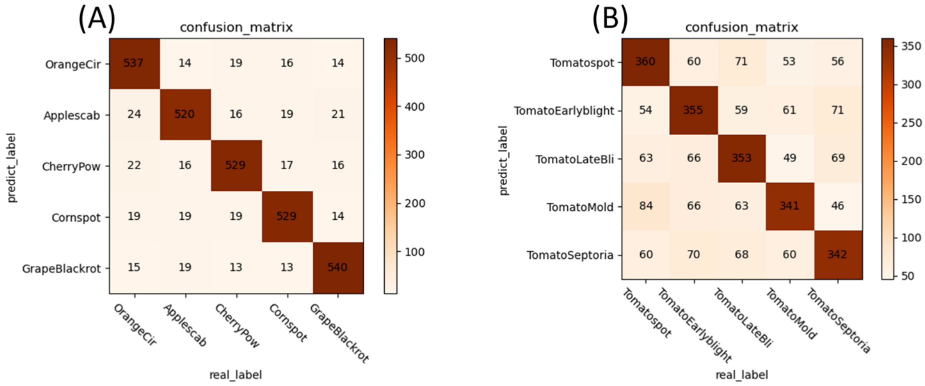

Figure 12. Fine-grained recognition confusion matrix. (**A**) Confusion matrix for recognition of different diseases of different crops. (**B**) Confusion matrix for recognition of different diseases of the same crop.

The rich cross-domain experimental results show that the model proposed in this study has strong cross-domain generalization capability. When migrating from the non-crop leaf disease domain to the crop leaf disease domain, as well as from laboratory scenes to complex field scenes, the performance of the model decreases to different degrees, but it is still better than the algorithms in Table 7. Especially in the few-shot field scenes where the training samples are extremely scarce, the model can still reach close to or even exceed the recognition effect of the traditional algorithm in the laboratory scenes, showing a strong adaptive ability. Meanwhile, this study also notes that increasing scene complexity and refining recognition granularity have a certain negative impact on the generalization performance of the model. This suggests that improving the generalization ability of the model in complex environments and fine-grained classification are key directions for intelligent recognition of crop leaf diseases from laboratory to field applications.

4. Discussion

4.1. The Impact of Way and Shot

N-way refers to the number of categories in the task, and K-shot refers to the number of support set images in the task. In order to explore the effect of shot and way on performance, this study counts the accuracy after changing the way and number of shots to quantify the impact. Figure 13A shows the variation of the accuracy with shot. The model prediction accuracy increases dramatically when the shots increase from 1 to 10. The accuracy increases from 91.20% to 98.29%. The change in accuracy stabilizes when the shot is greater than 10. In addition, the 95%MSE (mean square error) also tends to decrease when increasing the number of shots. (The trend of variance change is shown in Figure 13B). The 95%MSE (mean square error) decreases drastically from 1 shot to 5 shots, and stabilizes after 10 shots, which proves that the model predictions are getting more and more stable. Similarly, this subsection reports experiments conducted on Field-PV-split1 in a complex scenario, as shown in Figure 13C. The accuracy increased from 73.80% to 92.40% when the number of shots increased from 1 to 7. The complex scenario at 7 shots is comparable to the lab scenario 1-shot accuracy. Figure 14 shows the trend of 95%MSE (mean square error) and accuracy with the number of ways. When ways increase from 2 to 8, the accuracy decreases from 94.93% to 80.41% in the laboratory scenario and from 87.04% to 57.32% in the field scenario. The accuracy decreases with way increase. This means that FSL still remains limited in a wide variety of application scenes. On the other hand, unlike shot, 95%MSE fluctuates only slightly with increasing way, meaning that the number of ways has a minor effect on the stability of the model predictions.

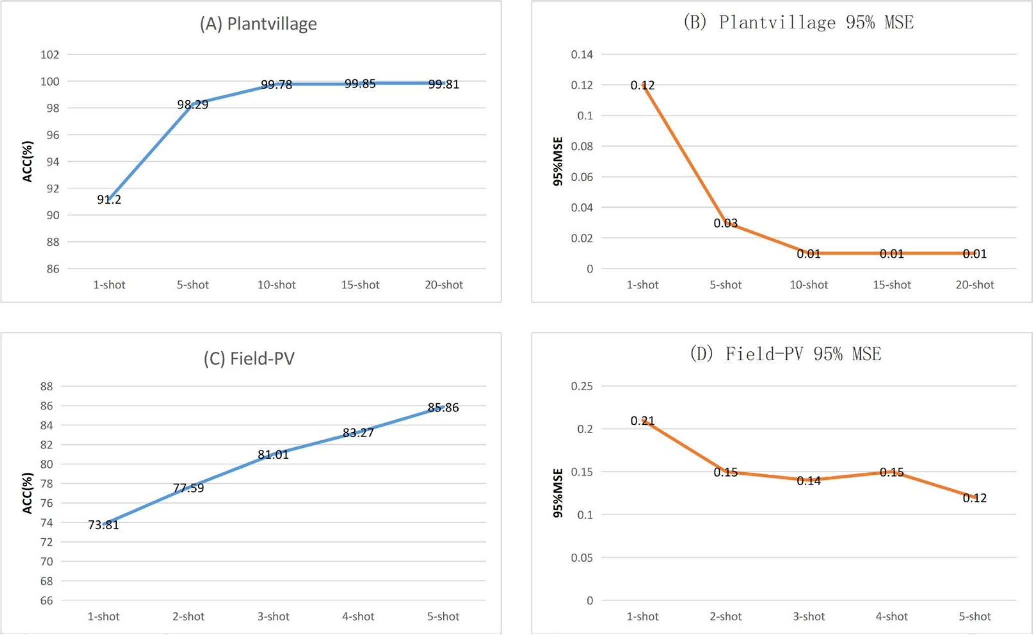

Figure 13. Model accuracy trend with shot. (**A**) Lab scene accuracy trend with shot. (**B**) Lab scene model stability (95%MSE) trend with shot. (**C**) Field scene accuracy trend with shot. (**D**) Field scene model stability (95%MSE) trend with shot.

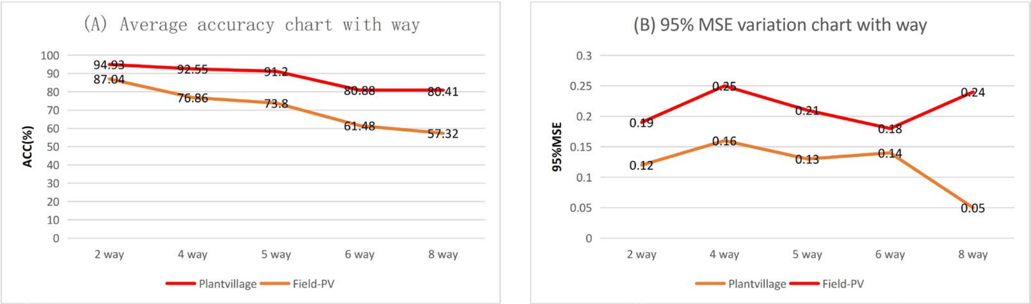

Figure 14. Model accuracy trend with way. (**A**) Accuracy trends in laboratory scenes and field scenes. (**B**) Stability (95%MSE) trends in laboratory scenes and field scenes.

The above phenomenon indicates that the accuracy increases with the increase of shot while the accuracy decreases with the increase of the number of ways; an increase in the number of ways will increase the information entropy, and more shots mean more a priori information. Therefore, in this study, we believe that when we face an application scenario with many categories, we should increase the number of support sets as much as possible as compensation to improve the accuracy.

4.2. Limitations and Future Work

Although the recognition method proposed in this study initially solves the problem of recognizing few shots in field scenes, the current method still has some limitations. First, each image in the Plantvillage and Field-PV datasets used in this study contains only one disease, but multiple diseases may occur on the same crop in the field scenes. This makes the model ineffective in recognizing multiple diseases on the same crop [48]. In future work, we intend to use multimodal information for semantic segmentation and feature calibration. Regarding fine-grained recognition, HMFN-FSL only achieves 58.13% recognition accuracy in field scenes. This limits the wide range of applications of the few-shot learning model for different diseases of the same crop and difficult disease recognition tasks. To address this issue, we intend to experimentally select a more robust feature extraction backbone to provide noise-resistant feature representations in future work. Finally, cross-domain few-shot learning is a promising approach for disease recognition. However, the typical scene is that when unpredictable external factors such as disease class, disease severity, weather, and light are present during the testing phase, the performance of the model degrades unimaginably. Therefore, we intend to address this issue from the perspective of the direction of training data collection and the choice of training strategy in order to narrow the gap and improve the performance of the model in future work.

5. Conclusions

Fine recognition of crop leaf diseases is an urgent need in the field of agricultural information. To overcome the difficulty of small-sample crop leaf disease recognition in field scenes, this study proposes a new few-shot learning network, HMFN-FSL, based on a meta-learning framework and an integrated learning approach, which improves the overall disease recognition accuracy as a whole. Taking the public dataset Plantvillage and the field scenes dataset Field-PV as the research object, the following conclusions are drawn by setting six training strategies to conduct training and testing experiments on the model:

(1) The impact of CBAM on the performance of base learner experiments shows that fusing the CBAM module into the feature extraction network of the base learner can significantly improve the feature extraction capability of the model. Compared to the base learner, the average accuracy of the embedded CBAM model increased by 2.14%.

(2) Compared to other base learners, the HMFN-FSL proposed in this study has higher accuracy and robustness. On 5way-1shot, the proposed model improves the accuracy of DeepEMD by 7.43% over the best base learners. The experimental results show that HMFN-FSL is effective for few-shot crop leaf disease recognition. Moreover, this study compares with state-of-the-art algorithms [32,33,36,42,46], and the model achieves the best performance. In addition, by changing the way and shot parameter configurations of the learning process, some key features affecting the classification accuracy were revealed. Overall, the model accuracy of the model increased as the number of shots increased, while the accuracy decreased as the number of ways increased.

(3) Cross-domain experiments of the model show that HMFN-FSL trained using the no-disease domain achieves 79.52% and 92.30% accuracy on the 5way-1shot and 5way-5shot tasks in the laboratory scenes, respectively. Moreover, it still shows high recognition accuracy on the complex scene dataset. The average recognition accuracy of the model reaches 73.80% and 85.86% on the 5way-1shot and 5way-5shot tasks, respectively. These results further demonstrate that the HMFN-FSL proposed in this study can be adapted to few-shot recognition in laboratory and field scenes.

In conclusion, the method proposed in this study provides an effective scheme for crop leaf disease recognition in few-shot field scenes and provides techniques and references for subsequent crop leaf disease recognition for few shots. In future developments, we intend to deploy it on robotic or automated devices to automatically monitor and recognize a wider range of plant disease information. Meanwhile, plant disease identification can be combined with other agricultural technologies, such as drones and sensor networks, to support precision agriculture.

Author Contributions: W.Y. (Wenbo Yan): Conceptualization, Formal analysis, Methodology, Visualization, and Writing—original draft preparation. Q.F.: Writing—review and editing, Funding acquisition, Validation, Resources, and Software. S.Y.: Data curation, Writing—review and editing, Funding acquisition, Validation, Resources, Software, and Project administration. J.Z.: Validation and Writing—review and editing. W.Y. (Wanxia Yang): Investigation and Software. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 32201663, 32160421, 31971792), funding from the Industrialization Support Project from Education Department of Gansu Province (Grant No. 2021CYZC-57).

Data Availability Statement: The PlantVillage dataset is publicly available online at https://github.com/spMohanty/PlantVillage-Dataset (accessed on 28 September 2018).

Acknowledgments: I would like to thank the technicians who assisted me with the data collection and analysis. Their support cannot be measured by value.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Oerke, E.C.; Dehne, H.W. Safeguarding production—Losses in major crops and the role of crop protection. *Crop Prot.* **2004**, *23*, 275–285. [CrossRef]
2. Strange, R.N.; Scott, P.R. Plant disease: A threat to global food security. *Annu. Rev. Phytopathol.* **2005**, *43*, 83–116. [CrossRef] [PubMed]
3. Lu, Y.; Yi, S.; Zeng, N.; Liu, Y.; Zhang, Y. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* **2017**, *267*, 378–384. [CrossRef]
4. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*, 2022. [CrossRef] [PubMed]
5. Ma, J.; Du, K.; Zheng, F.; Zhang, L.; Gong, Z.; Sun, Z. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Comput. Electron. Agric.* **2018**, *154*, 18–24. [CrossRef]
6. Long, M.; Ouyang, C.; Liu, H.; Fu, Q. Oil tea disease image recognition based on convolutional neural network and transfer learning. *Trans. Chin. Soc. Agric. Eng.* **2018**, *34*, 196–201. [CrossRef]
7. Zhang, X.; Qiao, Y.; Meng, F.; Fan, C.; Zhang, M. Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks. *IEEE Access* **2018**, *6*, 30370–30377. [CrossRef]
8. Dhiman, P.; Kaur, A.; Balasaraswathi, V.R.; Gulzar, Y.; Alwan, A.A.; Hamid, Y. Image Acquisition, Preprocessing and Classification of Citrus Fruit Diseases: A Systematic Literature Review. *Sustainability* **2023**, *15*, 9643. [CrossRef]
9. Gulzar, Y.; Ünal, Z.; Aktas, H.; Mir, M.S. Harnessing the Power of Transfer Learning in Sunflower Disease Detection: A Comparative Study. *Agriculture* **2023**, *13*, 1479. [CrossRef]
10. Hu, G.S.; Wu, H.Y.; Zhang, Y.; Wan, M.Z. A low shot learning method for tea leaf's disease identification. *Comput. Electron. Agric.* **2019**, *163*, 6. [CrossRef]
11. Chen, Y.P.; Pan, J.C.; Wu, Q.F. Apple leaf disease identification via improved CycleGAN and convolutional neural network. *Soft Comput.* **2023**, *27*, 9773–9786. [CrossRef]
12. Cap, Q.H.; Uga, H.; Kagiwada, S.; Iyatomi, H. LeafGAN: An Effective Data Augmentation Method for Practical Plant Disease Diagnosis. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 1258–1267. [CrossRef]
13. Gulzar, Y.; Hamid, Y.; Soomro, A.B.; Alwan, A.A.; Journaux, L. A Convolution Neural Network-Based Seed Classification System. *Symmetry* **2020**, *12*, 18. [CrossRef]
14. Mamat, N.; Othman, M.F.; Abdulghafor, R.; Alwan, A.A.; Gulzar, Y. Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach. *Sustainability* **2023**, *15*, 901. [CrossRef]
15. Zhang, G.; Li, Z.; Liu, H.; Liu, W.; Long, C.; Huang, C. Based on improved DenseNet and A Transfer Learning Model for Identifying Lotus Leaf Diseases and Pests. *Trans. Chin. Soc. Agric. Eng.* **2023**, *39*, 188–196. [CrossRef]
16. Li, Z.; Xu, J. Small sample recognition method of tea disease based on improved DenseNet. *Trans. Chin. Soc. Agric. Eng.* **2022**, *38*, 182–190. [CrossRef]
17. Yang, M.; Zhang, Y. Corn disease recognition based on the Convolutional Neural Network with a small sampling size. *Chin. J. Eco-Agric.* **2020**, *28*, 1924–1931. [CrossRef]
18. Gulzar, Y. Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. *Sustainability* **2023**, *15*, 1906. [CrossRef]
19. Mensink, T.; Verbeek, J.; Perronnin, F.; Csurka, G. Distance-Based Image Classification: Generalizing to New Classes at Near-Zero Cost. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2624–2637. [CrossRef]
20. Pan, S.J.; Yang, Q.A. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
21. Xiao, W.; Quan, F. Research on plant disease identification based on few-shot learning. *J. Chin. Agric. Mech.* **2021**, *42*, 138–143.
22. Lin, H.; Tse, R.; Tang, S.K.; Qiang, Z.P.; Pau, G. Few-shot learning approach with multi-scale feature fusion and attention for plant disease recognition. *Front. Plant Sci.* **2022**, *13*, 907916. [CrossRef] [PubMed]
23. Yang, J.C.; Guo, X.L.; Li, Y.; Marinello, F.; Ercisli, S.; Zhang, Z. A survey of few-shot learning in smart agriculture: Developments, applications, and challenges. *Plant Methods* **2022**, *18*, 12. [CrossRef] [PubMed]
24. Woo, S.; Park, J. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521v2.
25. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.H.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
26. Hughes, D.; Salathé, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060.
27. Gui, P.H.; Dang, W.J.; Zhu, F.Y.; Zhao, Q.J. Towards automatic field plant disease recognition. *Comput. Electron. Agric.* **2021**, *191*, 10. [CrossRef]

28. Sun, J.Q.; Cao, W.; Fu, X.; Ochi, S.; Yamanaka, T. Few-shot learning for plant disease recognition: A review. *Agron. J.* **2023**, *13*, 28. [CrossRef]
29. Abade, A.; Ferreira, P.A.; Vidal, F.D. Plant diseases recognition on images using convolutional neural networks: A systematic review. *Comput. Electron. Agric.* **2021**, *185*, 31. [CrossRef]
30. Wang, X.T.; Cao, W.Q. GACN: Generative Adversarial Classified Network for Balancing Plant Disease Dataset and Plant Disease Recognition. *Sensors* **2023**, *23*, 6844. [CrossRef]
31. Ghofrani, A.; Toroghi, R.M. Knowledge distillation in plant disease recognition. *Neural Comput. Appl.* **2022**, *34*, 14287–14296. [CrossRef]
32. Snell, J.; Swersky, K. Prototypical Networks for Few-shot Learning. *arXiv* **2017**, arXiv:1703.05175v2.
33. Vinyals, O.; Blundell, C. Matching Networks for One Shot Learning. *arXiv* **2017**, arXiv:1606.04080v2.
34. Greff, K.; Srivastava, R.K.; Koutnik, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [CrossRef] [PubMed]
35. Barratt, S. On the differentiability of the solution to convex optimization problems. *arXiv* **2018**, arXiv:1804.05098.
36. Zhang, C.; Cai, Y. DeepEMD: Differentiable Earth Mover's Distance for Few-Shot Learning. *arXiv* **2023**, arXiv:2003.06777v5. [CrossRef] [PubMed]
37. Chen, X.H.; Ji, A.M.; Cheng, G. A Novel Deep Feature Learning Method Based on the Fused-Stacked AEs for Planetary Gear Fault Diagnosis. *Energies* **2019**, *12*, 4522. [CrossRef]
38. Sivari, E.; Bostanci, E.; Guzel, M.S.; Acici, K.; Asuroglu, T.; Ayyildiz, T.E. A New Approach for Gastrointestinal Tract Findings Detection and Classification: Deep Learning-Based Hybrid Stacking Ensemble Models. *Diagnostics* **2023**, *13*, 720. [CrossRef]
39. Zhong, G.Q.; Zhang, K.; Wei, H.X.; Zheng, Y.C.; Dong, J.Y. Marginal Deep Architecture: Stacking Feature Learning Modules to Build Deep Learning Models. *IEEE Access* **2019**, *7*, 30220–30233. [CrossRef]
40. Wu, W.; Jing, X.Y.; Du, W.C.; Chen, G.L. Learning dynamics of gradient descent optimization in deep neural networks. *Sci. China-Inf. Sci.* **2021**, *64*, 15. [CrossRef]
41. Lee, D.G.; Jung, Y. Tutorial and applications of convolutional neural network models in image classification. *J. Korean Data Inf. Sci. Sociaty* **2022**, *33*, 533–549. [CrossRef]
42. Finn, C.; Abbeel, P. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv* **2017**, arXiv:1703.03400v3.
43. Xu, H.H.; Li, W.; Cai, Z.P. Analysis on methods to effectively improve transfer learning performance. *Theor. Comput. Sci.* **2023**, *940*, 90–107. [CrossRef]
44. Zhuang, F.Z.; Qi, Z.Y.; Duan, K.Y.; Xi, D.B.; Zhu, Y.C.; Zhu, H.S.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2021**, *109*, 43–76. [CrossRef]
45. Lu, J.; Behbood, V.; Hao, P.; Zuo, H.; Xue, S.; Zhang, G.Q. Transfer learning using computational intelligence: A survey. *Knowl. -Based Syst.* **2015**, *80*, 14–23. [CrossRef]
46. Hu, H.; Zhan, D.-C. Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
47. Wei, X.S.; Song, Y.Z.; Mac Aodha, O.; Wu, J.X.; Peng, Y.X.; Tang, J.H.; Yang, J.; Belongie, S. Fine-Grained Image Analysis with Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 8927–8948. [CrossRef] [PubMed]
48. Liu, J.; Wang, X.W. Plant diseases and pests detection based on deep learning: A review. *Plant Methods* **2021**, *17*, 18. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.