A prompt based approach for apple disease classification

xing sheng
Shandong Normal University
Yangyang Zhang
Shandong Normal University
Chen Lyu (lvchen@sdnu.edu.cn)
Shandong Normal University https://orcid.org/0000-0002-5044-1459

Research Article

Keywords: prompt, Disease classification, Transfer Learning, WPM, 0-shot

Posted Date: February 21st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1308089/v1

RESEARCH

A prompt based approach for apple disease classification

Xing Sheng, Yangyang Zhang and Chen Lyu*

*Correspondence:
lvchen@sdnu.edu.cn
School of Information Science and Engineering, Shandong Normal University, Jinan, China
Full list of author information is available at the end of the article

Abstract

Background: Apples occupy a large part of agricultural production as a fruit with a high yield and also a high nutritional value. However, diseases of the fruit and leaves of apples seriously affect the quality and yield of apples. In the past, people had to rely on their own experience to control apple diseases, however, this approach was poorly accurate, inefficient and did not meet the requirements of fruit farmers. Many current methods are based on convolutional neural networks, but convolutional neural networks usually require a large amount of labelled data to train the network, and datasets in the agricultural field can hardly meet this requirement.

Results: To solve this problem, this paper introduces zero-times learning, which can achieve equally good results even if the test object is a dataset that has never been seen before. Specifically, we give a short description of an image as a prompt word according to its category in the public dataset, form a graphical pair of the image to be trained and the corresponding prompt word, feed them into a deep convolutional neural network, and then pre-train it on a large public dataset, and migrate it directly to our dataset after the training is completed, saving a lot of resources. In addition, we also propose a new attention module WPM (Weighted-Pooling Module) to deeply mine feature vectors by combining weighted pooling operations with fully connected operations and activation functions. Through extensive experiments, we validate the effectiveness of the proposed approach of combining zero-times learning with prompt words and achieve good results on our own collected field dataset.

Conclusions: Our work provides new ideas and resource savings for disease classification tasks in agriculture.

Keywords: prompt; Disease Classification; Transfer Learning; WPM; 0-shot

Introduction

The apple is one of the four most important fruits in the world and the most important deciduous fruit tree in China; It ranks first in the country in terms of area and production. In Shandong province alone, the apple area has reached 800,000 hectares with a production of over 4 million tonnes, surpassing that of the USA and the Southern Hemisphere. However, it has been reported that there are as many as 100 disease problems of apples, which can be divided into leaf, branch, fruit and root diseases according to their location, the most serious and common of which are diseases of the leaves and fruit of apples. Only apple fruit ring rot and fruit anthracnose, the number of apples lost each year is above 5%, and in serious years it is as high as 60%.

In the past, the identification of diseases of apple fruit and leaves relied heavily on farmers conducting field visits and judging the type of disease based on experience. However, the lack of appropriate knowledge of some diseases among farmers has led to poor identification and assessment of the type

and severity of apple diseases and vague diagnostic criteria for diseases, resulting in failure to control diseases in a timely and reasonable manner. It is therefore important to identify and control apple diseases effectively, rationally and accurately, and to research and apply modern identification techniques. The use of computer vision-based technology for disease identification is a major way of achieving this goal. Not only can computer vision technology quickly and accurately obtain information about apple diseases, but it can also select the appropriate control method according to the severity of the disease. This will greatly save manpower and material resources, improve the efficiency of production and reduce costs.

Machine learning based methods have been commonly used for disease identification of fruits. Mohan et al. [1] used KNN and SVM to classify brown spot disease, leaf blast disease and bacterial blight disease of paddy plants with good results. Mokhtar et al. [2] used wavelet transform technique in combination with support vector machine and alternating kernel function to detect and identify diseases of tomato leaves and finally achieved 99.5% accuracy. Sindhuja et al. [3] used principal component analysis (PCA) on citrus yellow dragon disease pretreatment dataset and then used linear discriminant analysis, quadratic discriminant analysis and K-nearest neighbor methods to model and classify the dataset, and finally obtained an overall accuracy of 98%. Arivazhagan et al. [4] first built a color transformation structure of HIS on the input RGB images, then used specific thresholds to mask and remove green pixels, then performed a segmentation process to count their texture information, and finally used SVM for classification, achieving an accuracy of 94% of accuracy, and experimental results on a database of approximately 500 plant leaves confirmed the robustness of the method. However, although the research on plant pest recognition based on traditional image processing techniques has achieved certain results and the recognition accuracy of diseases is high, there are also shortcomings and limitations: the research process is tedious, relies too much on manually designed feature extraction methods, is highly subjective and time-consuming, etc. It cannot be adapted to practical application scenarios with more complex backgrounds and cannot meet the complex situations in practical applications, etc.

With the continuous development of deep learning, more and more research has applied deep learning to agricultural disease detection. Oppenheim et al. [5] collected 400 potato photos of different sizes, shapes and tones under different lighting conditions indoors, and by adding several new dropout layers behind the VGG [6] network to deal with the overfitting problem, they finally obtained the best Yusuke [7] proposed a convolutional neural network-based plant disease detection system using 800 cucumber leaf images taken in the field to train the convolutional neural network for detecting disease infection in two cucumber plants, and eventually achieved an average accuracy of 94.9% under a 4-fold cross-validation strategy. Fuentes et al. [8] proposed a deep learning-based approach to detect diseases in tomato plants using images taken by cameras of different resolutions, combining detectors such as Faster R-CNN with deep feature extractors (VGG and RESNet [9]) and proposing a method based on local and global class annotation and data enhancement to improve accuracy and reduce the number of false positives during training, ultimately achieving good results. Ferentinos et al. [10] developed a specialized deep learning model based on a specific convolutional neural network architecture and tested it on a publicly available dataset (87,848 images in the dataset, with photos taken both in controlled laboratory conditions and in the field), achieving 99.53% accuracy. Liu et al. [11] proposed a deep learning model based on two networks, VGG16 and ResNet50, to identify the species of large chrysanthemums, trained on a balanced dataset constructed from 14,000 images

of 103 cultivars, and ultimately achieved a top-5 accuracy of 98%. Although the convolutional neural network-based approach achieved very good results, the dataset and the time and resources required to train the convolutional neural network were too large for a plant disease dataset in agriculture to meet the requirements of a deep convolutional neural network.

To address these issues, in this paper we introduce zero-times learning. The basic idea of zero-times learning is to give machines the ability to reason so that the models we train can classify models that have never been seen before, achieving true "artificial intelligence". The definition of zero learning can be expressed as follows: given a labelled training instance D belonging to a seen category C, the goal of zero learning is to learn a classifier f(-):X→U that classifies a test instance X into a category U that it has not seen before. The label spaces covered by the training and test instances are disjoint. Therefore, zero-times learning is a sub-domain of migration learning [12]. In transfer learning, the source domain and the knowledge contained in the source task are transferred to the target domain to learn the model in the task goal. According to [12, 13], transfer learning can be classified as homogeneous transfer learning and heterogeneous transfer learning depending on whether the feature space and label space in the source and target domains/tasks are the same. In zero-times learning, the original label space is the visible class set, while the target label space is the invisible class set. Therefore, zero-times learning belongs to heterogeneous transfer learning. Since there are no labelled instances available in the invisible class set, some auxiliary information is needed to solve this problem. This auxiliary information should contain information about all invisible classes, which is also to ensure that the corresponding auxiliary information is provided for each invisible class. At the same time, the auxiliary information should be related to the instances in the feature space, this is to ensure that the auxiliary information is available.

In the existing work, the approach to incorporating supporting information is influenced by the way humans learn about the world. Humans can learn zero times with the help of some semantic background knowledge. For example, with the a priori knowledge that "wolves look like dogs, but their tails are short and thick and often hang back between their hind limbs", we can recognize a wolf even if we have not seen one, provided we know what a dog looks like and what a dog's tail looks like. The available auxiliary information is usually semantic, containing visible classes and some invisible classes. The approach taken in this paper is based on such semantic information in the form of prompt words. In particular, it should be noted that our approach differs from [14] in that the textual content in the method proposed in [14] for combining with text is described in detail and is far removed from our prompt words, and that it processes text and images separately using two models, only putting them together when the final output is produced. In contrast, our approach is to input the prompts and images into the model together as an image-text pair during training.

Material and methods

Image acquisition and material

The training set in the dataset we use is a public dataset and the test set is our own collected dataset. The training sets include ImageNet [15], PlantVillage [16] and PlantDoc [17]. The ImageNet dataset started in 2009 and was created by Professor Feifei Li and others. It has a total of 14,197,122 images and a total of 21,841 categories, with large categories including animals, birds, machines, flowers, food, fruits and 1000 other species, with roughly 1000 images per category. PlantVillage is a publicly

available dataset for testing machine learning plant disease detection algorithms. It contains 38 crop-disease pairs of 14 plant species, with a total of 54,299 images. All images were taken indoors with a single background image. The PlantDoc dataset is a dataset of crop disease images, manually annotated with images acquired online, and includes 27 disease categories (10 healthy types, 17 disease types) for 13 plants, with a total of 2598 images for image classification and target detection. The provider of the data for our test set is the Shandong Academy of Agricultural Sciences in Jinan, Shandong Province, China. The test set contains a total of 1204 images of apple leaves and fruits, including 474 images of apple fruits and 730 images of apple leaves. The images were taken with a mobile phone under real production conditions, at a resolution of 3456*4608, with non-uniform light intensity (none of the images were taken with the flash on), at different angles and distances, etc. The images contain a lot of extraneous background information such as other apples, leaves, branches and sky. The dataset used in this paper includes normal leaves and fruits, of which there are six types, all in .jpg format. As shown in Table 1, there are three types of apple fruit: healthy fruit, apple fruit ring rot and apple fruit anthracnose, and three types of apple leaves: healthy leaves, apple anthracnose leaf blight and apple leaf rust. Some of the images in the dataset are shown in Figure 1.

Dataset.pdf

Figure 1: Image of part of the data in the dataset.

The overall flow of our method is shown in Figure 2. First, we collect disease samples of leaves and fruits of apples in orchards under the guidance of an expert to label and classify them. The collected images were then subjected to a series of processes, such as image cropping, image contrast stretching, grey level slicing, dynamic range compression, etc. They were set to a uniform size of 224*224 during training, and then the images were subjected to normalization operations, etc. A prompt word was then added to each category, such as “A photo of a healthy apple.”, and placed together with the corresponding image as a picture-text pair, trained in the same way as Clip. Once the training was completed, the weights were tested directly on our own dataset without any changes to the weights file. Inspired by [25], we trained the model under a total of two prompt words: “A photo of a {label}” and “A photo of a {label}, a type of XX. We trained on three common datasets: ImageNet, PlantVillage, and PlantDoc. Table S1 and S2 show the model trained on “A photo of a {label}” and “A photo of a {label}, a type of XX.” prompt. In particular, because the image scenes in ImageNet are complex and cannot be described simply by the two prompt words mentioned

Table 1: Apple fruit and leaf dataset display.

ClassID	Type	Number of Sample
0	Healthy Apple	157
1	Ring Rot Apple	172
2	Anthrancnose Apple	145
3	Healthy Leaf	67
4	Anthrancnose Leaf	435
5	Rust Leaf	228

above, in practice we use more complex prompt words, such as “A dark photo of a {}.” or “A black and white photo of a {}.” . The model was trained and tested on a Linux server with an Intel(R) Core(TM) i9-10980XE CPU (128GB RAM) and accelerated with two Nvidia GeForce 2080Ti GPUs (24G RAM). The model was implemented in the pytorch 1.8 open source framework with python version 3.7. The learning rate was set to 4e-6,eps to 1e-6 for Clip-RN50 and 5e-6,eps to 2e-6 for both Clip-ViT-B/16 and Clip-ViT-B/32, and the weight decay was 0.1, and the optimization functions used were all AdamW.

FrameWork.pdf

Figure 2: Overall flow chart of the method.

Evaluation Metric

We use accuracy, precision, recall and F1-score as our evaluation metrics.

Accuracy.

Accuracy refers to the number of correctly classified samples as a proportion of the number of samples determined to be positive by the classifier, and is often used to evaluate the quality of the results, which can be expressed as:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Where TP represents true cases, FP represents false positive cases, TN represents true negative cases and FN represents false negative cases.

Precision.

Precision is often used to evaluate the completeness of the results and it can be expressed as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall.

Recall refers to the proportion of correctly classified positive samples to true positive samples and is often used to indicate coverage and can be expressed as:

$$\text{Recall} = \frac{Tp}{TP + FN} \tag{3}$$

F1-score. The F1-score is a weighted summed average of precision and recall proposed to balance precision and recall, which is as suggestive of precision and recall as possible while expecting the difference between them to be as small as possible, and can be expressed as:

$$\text{F1} = 2\frac{P * R}{P + R} \tag{4}$$

Prompt

With the continued research and development of deep convolutional networks, the learning approach to natural language processing has begun to gradually shift from a fully supervised approach to a pre-trained-fine-tuned model [18, 19, 20], in which models with fixed architectures are pre-trained as language models that predict the probability of observed textual data. Due to the very large and rich amount of raw text on which the language model is trained, the model can learn powerful generic features of the language it models during the training process. The trained language model is then applied to downstream tasks by fine-tuning it with additional parameters and using task-specific objective functions. In this model, the focus shifts to goal engineering, designing training goals for the pre-training and fine-tuning phases, which can lead to better pre-trained models for text summary pre-training [21]. And by now (2022), the pre-training-fine-tuning model is being replaced by a pre-training-prompt word-fine-tuning model. In this model, instead of adapting the pre-trained language model to downstream tasks through goal engineering, the downstream task is reformulated so that it solves these downstream tasks in the original language model with the help of textual prompts. For example, when completing a fill-in-the-blank task, "I ate a fruit today", we can prompt "it was very __" and have the language model fill in the blank with an adjective. With appropriate prompts, we can get the desired output from the pre-trained language model [22, 23, 24]. This model has also been applied to the image domain.

Clip.pdf

Figure 3: Clip's overall framework diagram.

Alec Radford et al. [25] proposed to use image-text pairs for pre-training (shown in Figure 3) and migrate the completed training model to a new task to achieve zero learning. Specifically: Firstly, a large number of unpurged pairs of text and image pairs are searched on the Internet, they are labelled and encoded with text and image respectively, and the similarity between two pairs of text and image in a batch is calculated by dot product to obtain a batch size by batch size similarity matrix, and the similarity on the diagonal is the similarity value of the positive sample. Therefore, in the training process, the optimization goal is to make the similarity value of the positive samples as large as possible. In the inference of the image classification task, the model first needs to convert the category labels into the same sentences as in the pre-training (where prompt is used), then get the prompt words corresponding to the different categories and then input them into the test network to form an image-text pair for zero-learning prediction, and output the corresponding categories after the prediction is completed. After it was proposed, many studies have applied Clip to the image and video domains, such as CoOp [26], Action Clip [27], Clip4Caption [28], Clip4Clip [29], etc.

Deep learning based classification framework

Based on ResNet

The proposed deep residual network is a milestone event in the history of convolutional neural network images, which solves the problem of difficult training of deep convolutional neural network models. Specifically, as the depth of the network increases, the network can perform more complex feature extraction operations and theoretically achieve better results, but in the actual training process, the deep network will suffer from degradation problems: the network accuracy saturates

or even decreases when the depth of the network increases. To solve this problem, He Keming et al. [9] proposed ResNet, which solves the degradation problem through residual learning. When the input is x its learned features are noted as H(x), and residual learning is performed by means of H(x) = F(x) + x (as shown in Figure 4).The ResNet network improves on the network of VGG19 by adding residual units through a short-circuiting mechanism, and achieves good results. One of the experimental models baseline we used was the RN50 model.

ResNet.pdf

Figure 4: Residual learning units.

Based on Vision Transformer

A detailed framework diagram of the Vision Transformer used in this paper is shown in Figure 5. Alexey et al. [30] first applied the Transformer to the field of computer vision and proposed the Vision Transformer model. Specifically, it first chunks the images and then spreads each image into a one-dimensional vector; next, a linear transformation (i.e., a fully connected layer) is done on each vector, called Patch Embedding. A vector is artificially added to the input vector as the final classification vector. They are then fed into the Transformer's encoder for mask computation using multi-headed self-attentiveness and feature mapping using an FFN (Feed-Forward Network), and the processed feature vectors are fed into a Multilayer Perceptron (MLP) for classification. We used the Vision Transformer trained ViT-B/16 and ViT-B/32 (B denotes Base, a relatively small amount of data; 16 and 32 denote the input patch size of 16*16 and 32*32 respectively) models in our baseline experiments.

ViT.pdf

Figure 5: Architecture diagram of Vision Transformer.

Weighted-pooling Module

As shown in Figure 6, a novel attention mechanism is proposed in order to reduce the number of parameters and to facilitate the modelling of the information in the feature vector. This attention mechanism uses a simple combination of a pooling layer with additional weights, a fully connected layer, a batch normalization (BN) layer and a Sigmoid activation function layer.

WPM.pdf

Figure 6: Proposed weighted-pooling Module.

Specifically, for the feature vector Feature, the overall algorithmic process can be divided into two parts: the first part is the feature extraction operation and the second part is the feature fusion operation. The feature extraction operation can be formulated as follows:

$$Feature_1 = S(\mathrm{ReLU}(FC((\mathrm{Cat}(\gamma\,\mathrm{Avg}(\text{ Feature }), \beta\,\mathrm{Std}(\mathrm{Feature})))))) \tag{5}$$

where Avg and Std denote AvgPool and StdPool operations respectively, Cat stands for connected operation and FC stands for fully connected operation, ReLU and S stand for ReLU activation function and Sigmoid function respectively, γ and β are two weights.

$$\text{Fused} = Feature_1 \otimes \text{Feature} \oplus \text{Feature} \tag{6}$$

where $\otimes$ and $\oplus$ represents the multiplication of elements and the addition of elements respectively, Fused denotes the feature vector of the total post-fusion.

Results and discussions

Comparison using different datasets. From Figure 7, we can see that ImageNet outperforms PlantVillage and PlantDoc in classifying the fruit of apples.

Results and reasons: This is because the ImageNet dataset has more diverse images, including a variety of fruits including apples, so it is better able to distinguish between the types of apples. The images in the PlantVillage and PlantDoc datasets are of plant leaves and lack images of fruit, so they are less generalizable to apple fruit.

Apple.pdf

Figure 7: Comparison of average accuracy on pre-trained models on different datasets for classification of apple fruits..

From Figure 8, we can see that the results on PlantDoc are much better than those on PlantVillage and ImageNet in terms of leaf classification of apples.

Leaf.pdf

Figure 8: Average accuracy comparison of pre-trained models on different datasets for the classification of apple leaves.

Results and reasons: This is due to the lack of plant leaf image data on ImageNet, whereas PlantDoc has leaf images of a wide range of plant diseases in the field, including apple leaf disease images, and is therefore better able to classify diseased apple leaves. As for PlantVillage, although it has a variety of plant disease foliage in its dataset, including apple disease foliage, its foliage was taken under controlled indoor conditions and lacks good generalization to images taken in the field.

Comparison of different prompt words. As shown in Figure 9, the performance of our two trained prompt words on our apple fruit and leaf datasets after pre-training on both datasets on the four metrics illustrates that "A photo of a {}, a type of {}." is better than "A photo of a {}." overall better performance. The quantitative analysis tables using the different prompt words are shown in Table S1 and S2.

Result and reason: "A photo of a {}, a type of {}". The phrase "a type of {}" contains more semantic information than the other prompt, which contains the category of the image being trained. This makes it easier for the model to combine the semantic information of the text with the feature information of the image, and the training results are better.

Prompt.pdf

Figure 9: Plot of the average accuracy of the two prompt words on the two datasets (PlantVillage and PlantDoc) compared.

Effectiveness of the prompt word method. As we can see in Figure 10, the models trained on ImageNet, PlantVillage and PlantDoc using the prompt word approach were tens of times more effective in identifying diseases on apple fruit and leaves than the approach without the prompt word. The quantitative analysis of the models without the use of prompt words is tabulated in Table S3.

Prompt Compare.pdf

Figure 10: Histogram of average accuracy on the three models with and without the prompt words.

Results and reasons: When training with prompt words, the model will first calculate the similarity between the prompt word and the image, and will get the image and text with the greatest similarity during training, and will be able to make good predictions when inferring with the semantic information in the text, whereas a directly pre-trained model without this semantic information from the text of the prompt word will be ineffective when faced with an image that it has not seen before.

Effectiveness of the proposed attention module. As we can see in Table S4, after using our proposed attention module, all of the pre-trained models showed varying degrees of improvement in the four evaluation metrics.

Results and reasons: The effectiveness of our proposed attention module is illustrated by its ability to better extract the associated features from the feature vector, improving its effectiveness in classification tasks.

Comparison of the validity of different models. As shown in Figure 11, we trained a total of three models, and from their comparison we can see that ViT-B/16 performed the best, ViT-B/32 the second best, and then RN50.

Model-Compare.pdf

Figure 11: Histogram comparing the average accuracy of each model tested on the Apple dataset.

Results and reasons: Firstly, the performance of the RN50-based model is less effective than that of the Vision Transformer-based model, because the Vision Transformer divides the image into multiple patches during training, which can learn the feature information between adjacent pixels and help the network to distinguish the spatial information in the image, while the CNN-based The CNN-based RN50 does not learn this information very well, and is therefore less effective. ViT-B/16 is more granular and learns more features than ViT-B/32, so it is more effective than ViT-B/32.

Table 2 shows the histogram analysis of the results of the three deep convolutional neural network models on the Apple dataset after pre-training on the three dataset pre-training models, and provides the mean μ and standard deviation δ of the four evaluation metrics on the Apple dataset.

Table 2: Box line plots of the four evaluation metrics (the four metrics in the box line plots, accuracy, recall, precision and f1-score, are indicated by red, orange, green and indigo blue respectively) on the ImageNet, PlantVillage and PlantDoc datasets experimented with the three models (RN50, ViT-16, ViT-32) and are given corresponding to the box line plots. The mean (μ) and standard deviation (δ) are given.

Models	Boxplot for Datasts			Region		Evaluation Metric			
	ImageNet	PlantVillage	PlantDoc			Accuracy	precision	recall	F1-score
RN50	RN50-ImageNet.pdf	RN50-PlantVillage.pdf	RN50-PlantDoc.pdf	L	μ	51.27	19.53	36.11	22.88
					δ	35.79	15.8	6.81	15.86
				D	μ	43.26	14.43	33.33	17.59
					δ	37.27	12.42	6.35	11.91
				S	μ	50.86	19.68	36.11	21.55
					δ	44.87	19.26	6.81	19.04
ViT-16	ViT-16-ImageNet.pdf	ViT-16-PlantVillage.pdf	ViT-16-PlantDoc.pdf	L	μ	54.98	24.89	38.89	29.39
					δ	28.93	14.58	8.61	13.7
				D	μ	66.12	25.32	38.89	28.29
					δ	37.14	14.55	8.61	13.7
				S	μ	59.71	24.97	33.89	28
					δ	36.34	18.37	8.61	17.98
ViT-32	ViT-32-ImageNet.pdf	ViT-32-PlantVillage.pdf	ViT-32-PlantDoc.pdf	L	μ	57.28	20.61	36.11	19.6
					δ	27.61	9.76	6.81	10.88
				D	μ	64.73	23.15	36.11	19.6
					δ	29.34	14.15	6.81	10.88
				S	μ	54.35	29.23	44.44	31.7
					δ	42.29	36.68	27.22	35.25

As can be seen from the 9 images (RN50-ImageNet.pdf, RN50-PlantVillage.pdf, RN50-PlantDoc.pdf,ViT-16-ImageNet.pdf, ViT-16-PlantVillage.pdf, ViT-16-PlantDoc.pdf, ViT-32-ImageNet.pdf, ViT-32-PlantVillage.pdf, ViT-32-PlantDoc.pdf), the range of accuracy of the test results on all three datasets is very wide, which indicates that the results of the prompt word method fluctuate widely across the images, suggesting that it is not stable, and the results are sometimes good and bad. In terms of stability, the RN50 network has the lowest outlier value, but it also has the smallest median value, indicating that it is not as sensitive to the feature information of the images in the test set as the other two networks and is less effective, which is consistent with the reasons analyzed in the previous section. Compared to ViT16, ViT32 has more outliers and the results vary more between the pre-trained models of different datasets, suggesting that ViT32 is more susceptible to the influence of the training images during training and lacks good generalization.

As for the recall, it is much more stable compared to the accuracy, but the mean and median values are significantly smaller than the accuracy, probably because the model did not learn enough feature information from the training dataset, a result that is consistent with the large gap between the pre-training and testing datasets we used.

The F1-score values used to balance accuracy and recall are more representative of the characteristics of the three models: RN50 has the smallest value while ViT-16 has a narrower F1-score width, indicating a smaller fluctuation range and a more stable effect and better generalization performance than ViT-32.

Conclusions

In this paper, we introduce prompt words into the agricultural disease classification task for the first time for zero learning in order to address the problem of small number of datasets and diffi-

cult collection in the agricultural domain. We used three deep convolutional neural network models: RN50, ViT-16 and ViT-32, using two prompt words: "A photo of a {}." and "A photo of a {}, a type of {}." Extensive experiments were done on three public datasets: ImageNet, PlantVillage and PlantDoc. Through these experiments, we verified the feasibility and effectiveness of the approach using prompt words (the approach using prompt words is tens of times better than training directly from public datasets such as ImageNet), and compared and analyzed the effectiveness of the two prompt words. In addition, in order to deeply mine the relationships between feature vectors and improve the effectiveness of deep convolutional neural networks, we also proposed a new attention module, WPM. Through a weighted pooling operation and some other operations, an average accuracy improvement of 5.2% on apple fruits and 5.7% on apple leaves was achieved. Our work provides new ideas and resource savings for disease classification tasks in agriculture.

Appendix

Table S1. Results of experiments with RN50 and Vision Transformer-based models on apple fruit and leaves, with the prompt "A photo of a {label}.".

Table S2. Results of experiments with RN50 and Vision Transformer-based models on apple fruit and leaves with the prompt "A photo of a {label}, a type of a .".

Table S3. Quantitative analysis of pre-trained models on our dataset without the use of prompt words.

Table S4. Quantitative analysis on our dataset using the pre-trained model after attention (prompt: "A photo of a .").

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Availability of data and materials
Not applicable.

Competing interests
The authors declare no competing interests.

Funding
Not applicable.

Authors' contributions
XS carried out the simulations and generated the results. XS and YZ curated and annotated the datasets. CL conceived, designed, and supervised the work. XS and YZ wrote the manuscript. CL supervised the annotation and classification of diseases. CL supervised the data management and testing.

Acknowledgements
Not applicable.

Authors' information
Shandong Normal University, Jinan, China.

Author details
School of Information Science and Engineering, Shandong Normal University, Jinan, China.

References

1. Mohan, K.J., Balasubramanian, M., Palanivel, S.: Detection and recognition of diseases from paddy plant leaf images. International Journal of Computer Applications **144**(12) (2016)
2. Mokhtar, U., Ali, M.A., Hassenian, A.E., Hefny, H.: Tomato leaves diseases detection approach based on support vector machines. In: 2015 11th International Computer Engineering Conference (ICENCO), pp. 246–250. IEEE

3. Sankaran, S., Mishra, A., Maja, J.M., Ehsani, R.: Visible-near infrared spectroscopy for detection of huanglongbing in citrus orchards. Computers and electronics in agriculture **77**(2), 127–134 (2011)
4. Arivazhagan, S., Shebiah, R.N., Ananthi, S., Varthini, S.V.: Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. Agricultural Engineering International: CIGR Journal **15**(1), 211–217 (2013)
5. Oppenheim, D., Shani, G.: Potato disease classification using convolution neural networks. Advances in Animal Biosciences **8**(2), 244–249 (2017)
6. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-scale Image Recognition. (2014)
7. Kawasaki, Y., Uga, H., Kagiwada, S., Iyatomi, H.: Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In: International Symposium on Visual Computing, pp. 638–645 (2015). Springer
8. 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 **17**(9), 2022 (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. Ferentinos, K.P.: Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture **145**, 311–318 (2018)
11. Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., Hughes, D.P.: Deep learning for image-based cassava disease detection. Frontiers in plant science **8**, 1852 (2017)
12. Torrey, L., Shavlik, J.: Transfer learning, pp. 242–264. IGI global, ??? (2010)
13. Day, O., Khoshgoftaar, T.M.: A survey on heterogeneous transfer learning. Journal of Big Data **4**(1), 1–42 (2017)
14. Wang, C., Zhou, J., Zhao, C., Li, J., Teng, G., Wu, H.: Few-shot vegetable disease recognition model based on image text collaborative representation learning. Computers and Electronics in Agriculture **184**, 106098 (2021)
15. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A Large-scale Hierarchical Image Database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). Ieee
16. Hughes, D., Salathé, M., et al.: An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060 (2015)
17. Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., Batra, N.: PlantDoc: a Dataset for Visual Plant Disease Detection. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, pp. 249–253 (2020)
18. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
19. Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., Hon, H.-W.: Unified language model pre-training for natural language understanding and generation. arXiv preprint arXiv:1905.03197 (2019)
20. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
21. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning, pp. 11328–11339 (2020). PMLR
22. Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.H., Riedel, S.: Language models as knowledge bases? arXiv preprint arXiv:1909.01066 (2019)
23. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
24. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
25. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., *et al.*: Learning Transferable Visual Models from Natural Language Supervision. (2021)
26. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. arXiv preprint arXiv:2109.01134 (2021)
27. Wang, M., Xing, J., Liu, Y.: Actionclip: A new paradigm for video action recognition. arXiv preprint arXiv:2109.08472 (2021)
28. Tang, M., Wang, Z., Liu, Z., Rao, F., Li, D., Li, X.: CLIP4Caption: CLIP for Video Caption. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 4858–4862 (2021)
29. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval. arXiv preprint arXiv:2104.08860 (2021)
30. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

Additional Files

Additional file 1 — Sample additional file title

Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title

Additional file descriptions text.

Figures

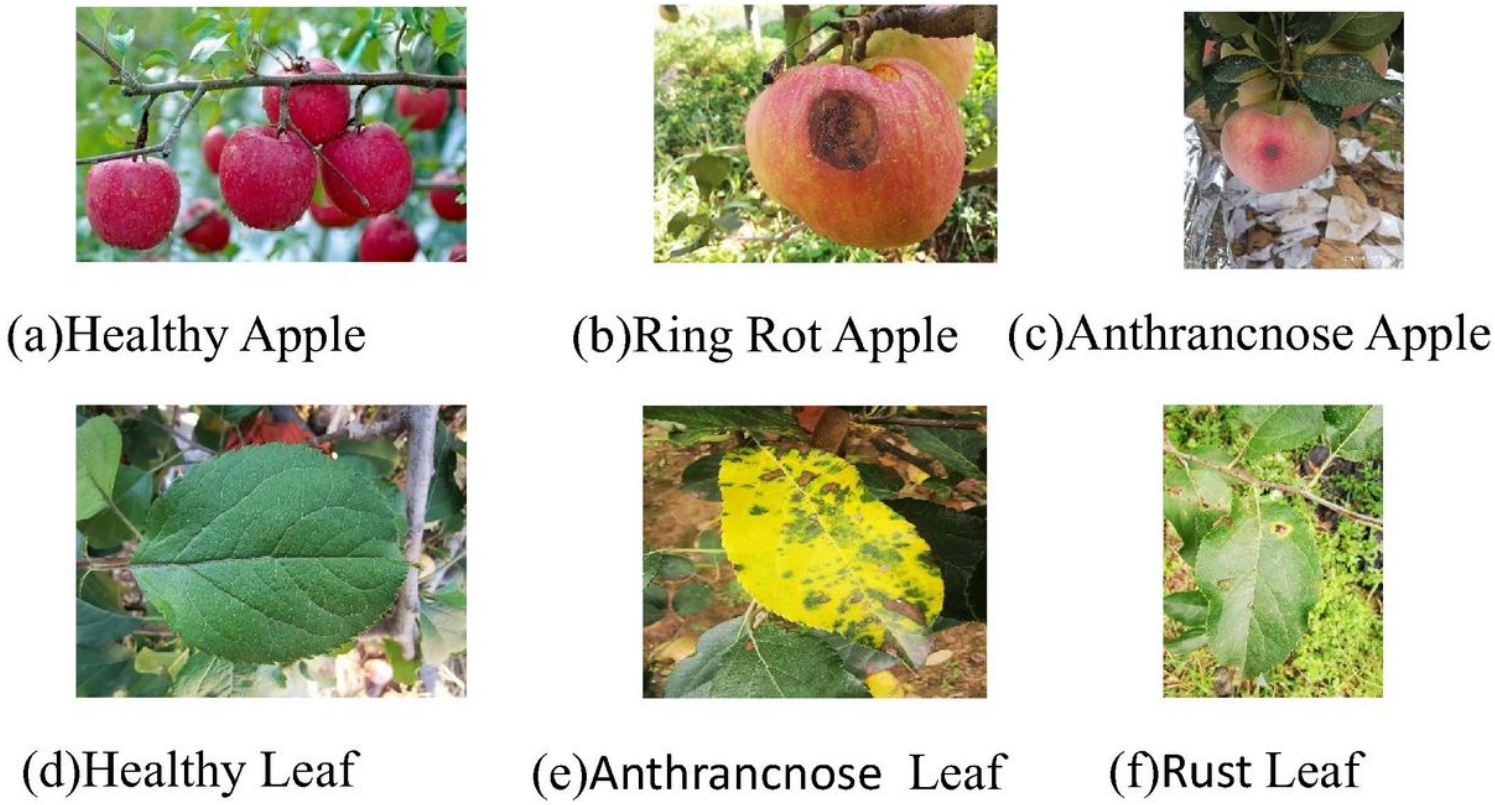

Figure 1

Image of part of the data in the dataset.

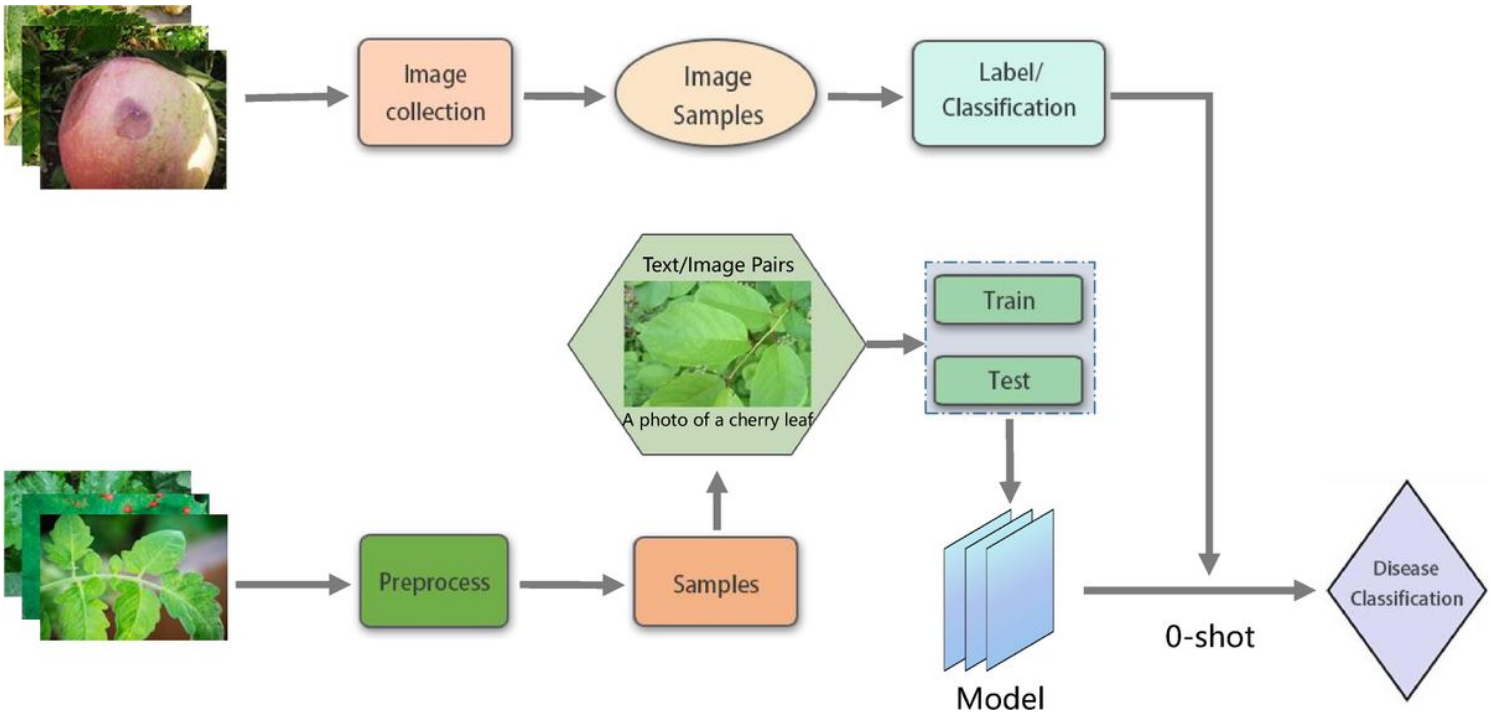

Figure 2

Overall flow chart of the method.

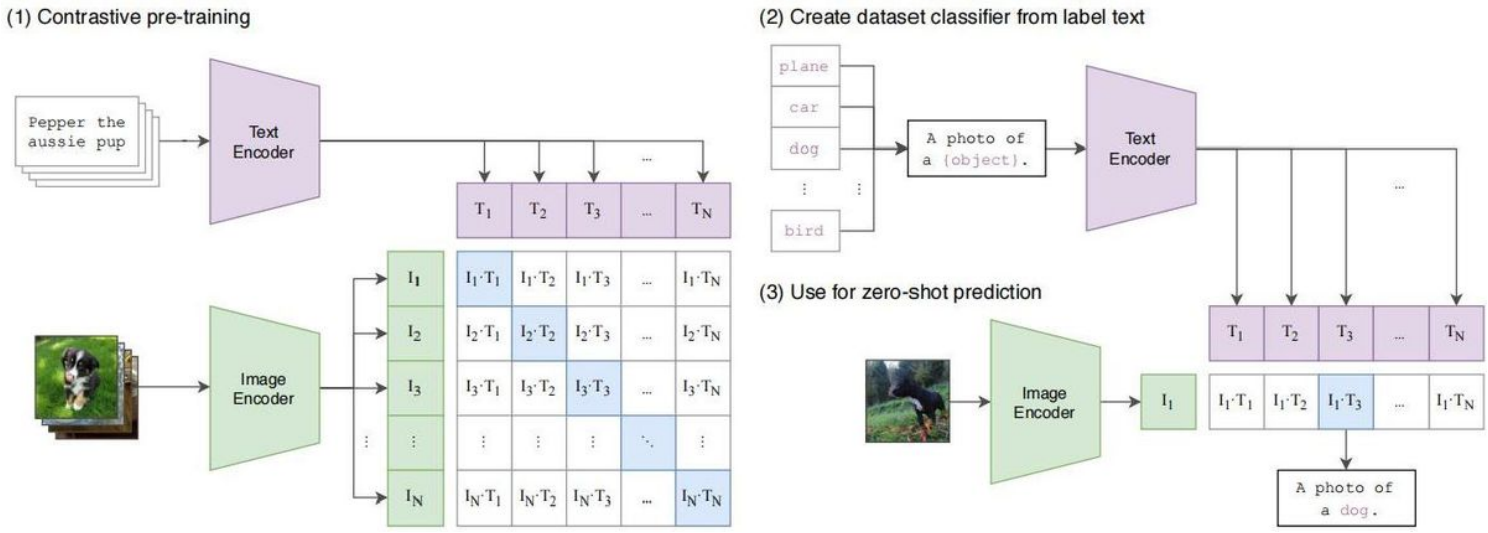

Figure 3

Clip's overall framework diagram.

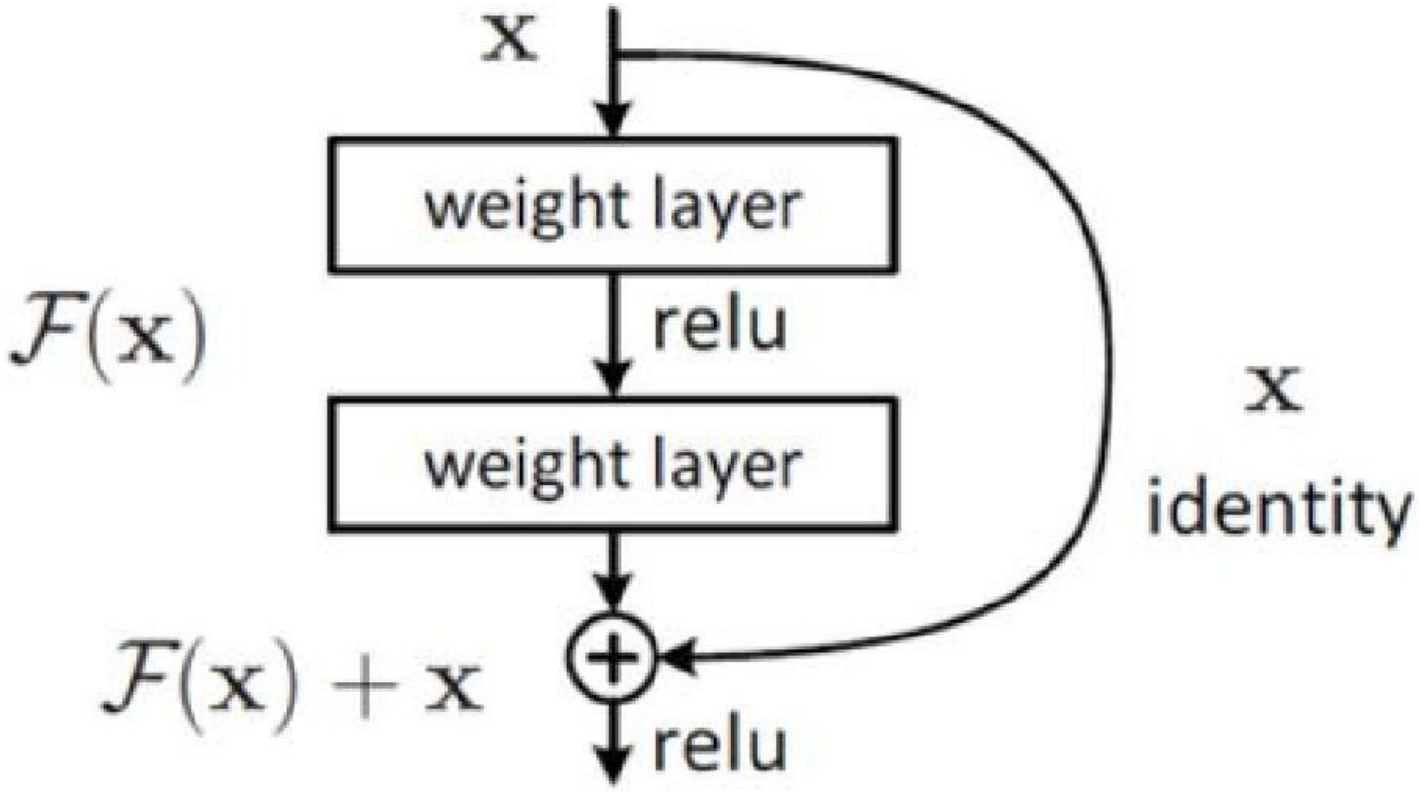

Figure 4

Residual learning units.

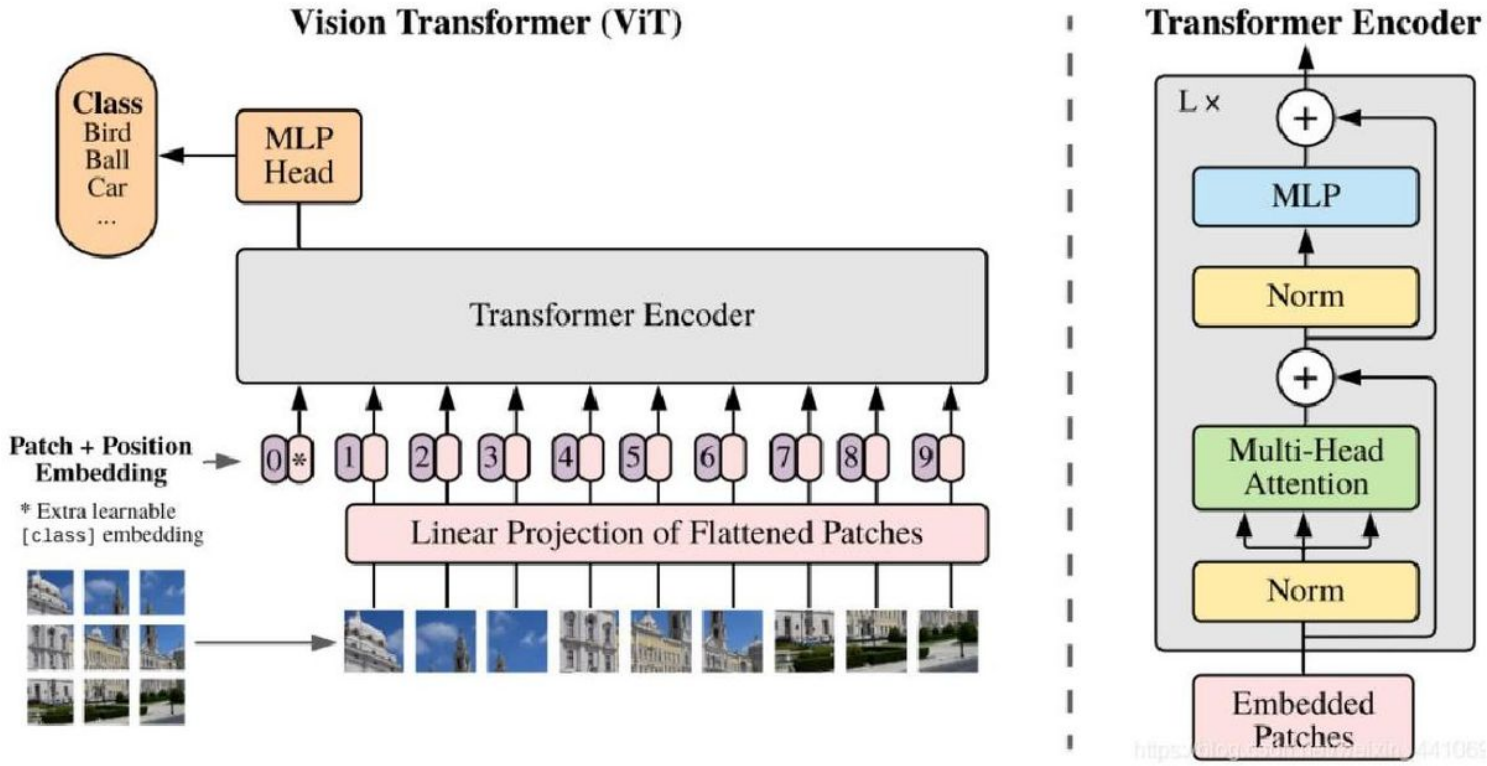

Figure 5

Architecture diagram of Vision Transformer.

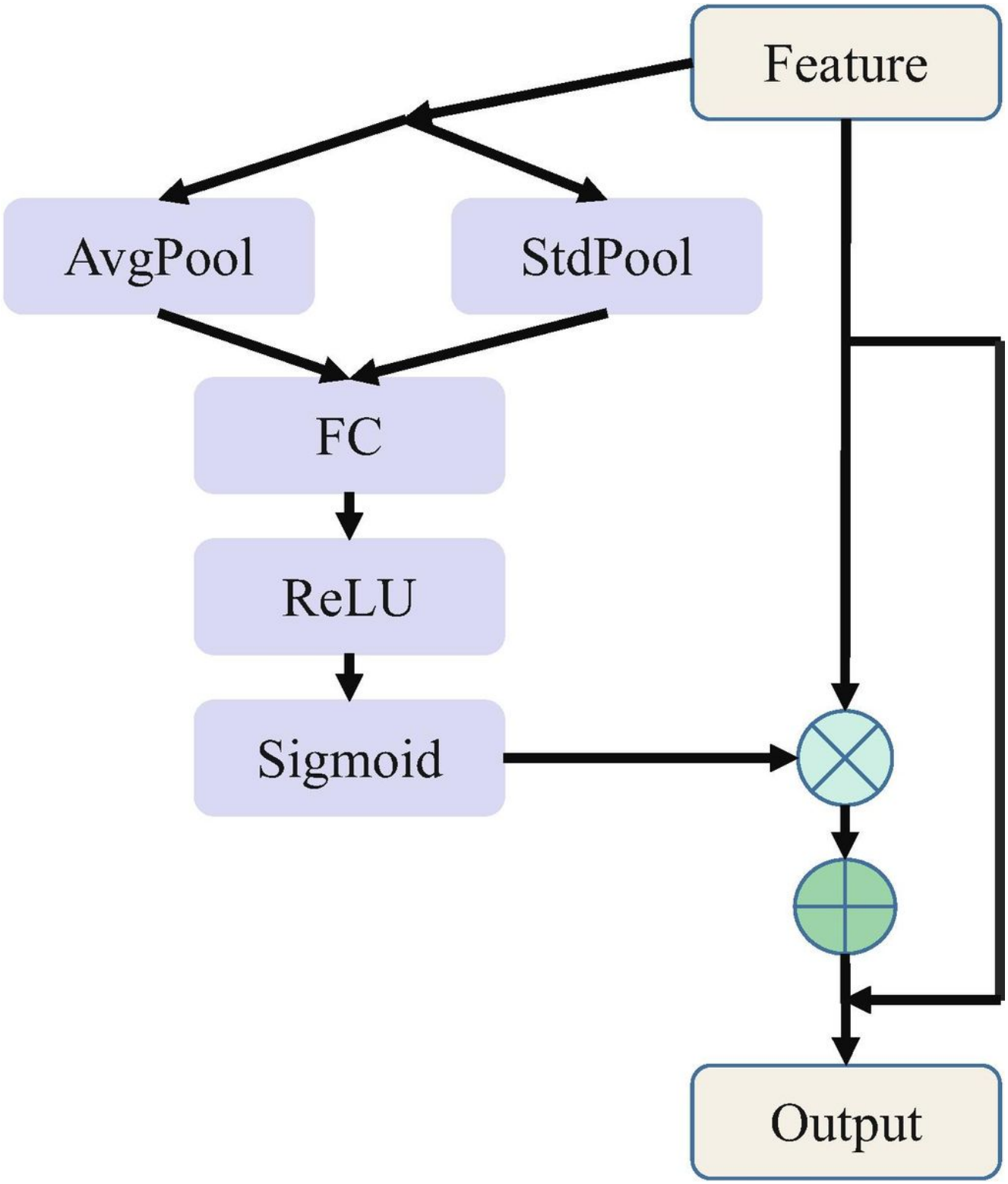

Figure 6

Proposed weighted-pooling Module.

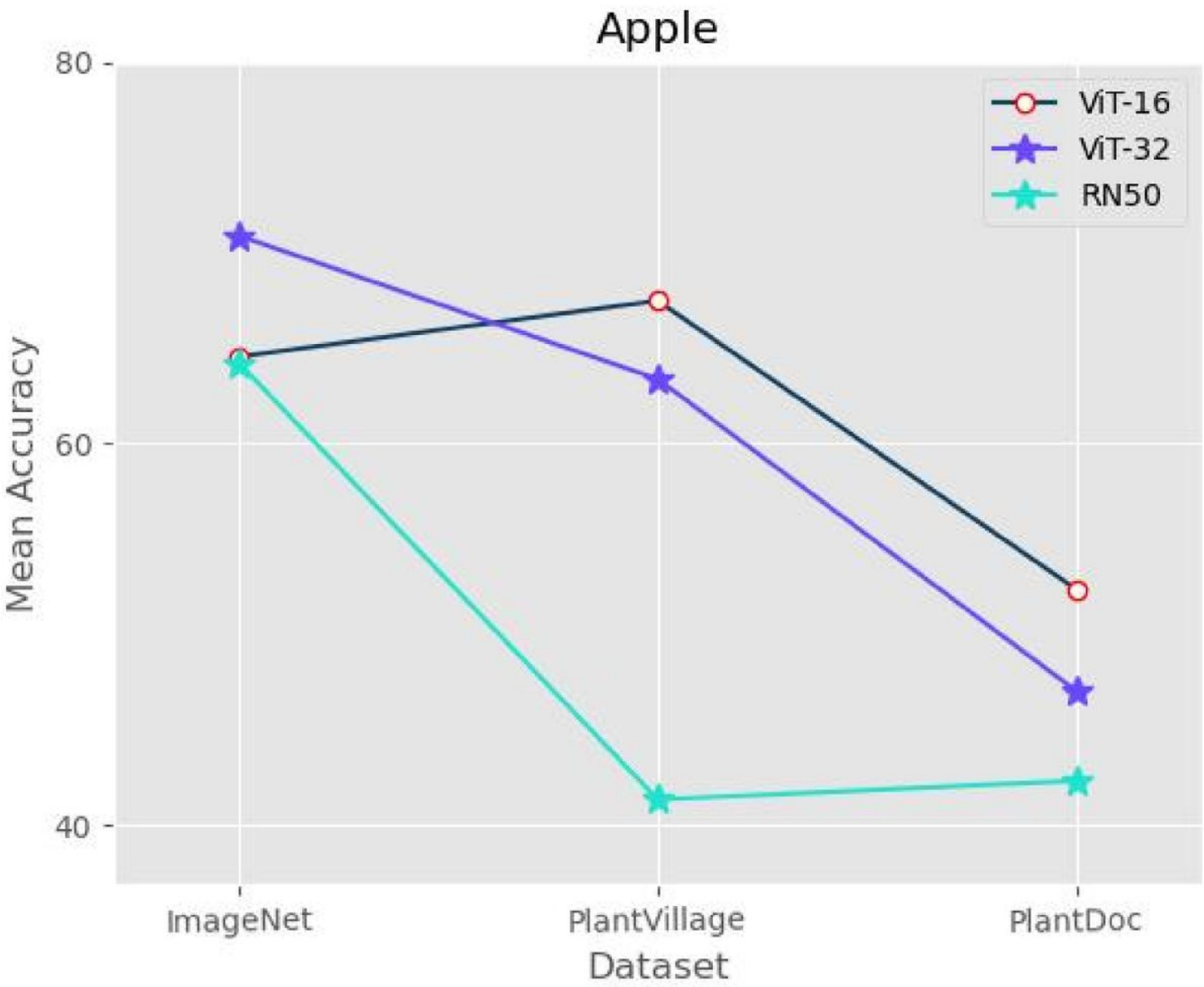

Figure 7

Comparison of average accuracy on pre-trained models on diferent datasets for classi-fication of apple fruits..

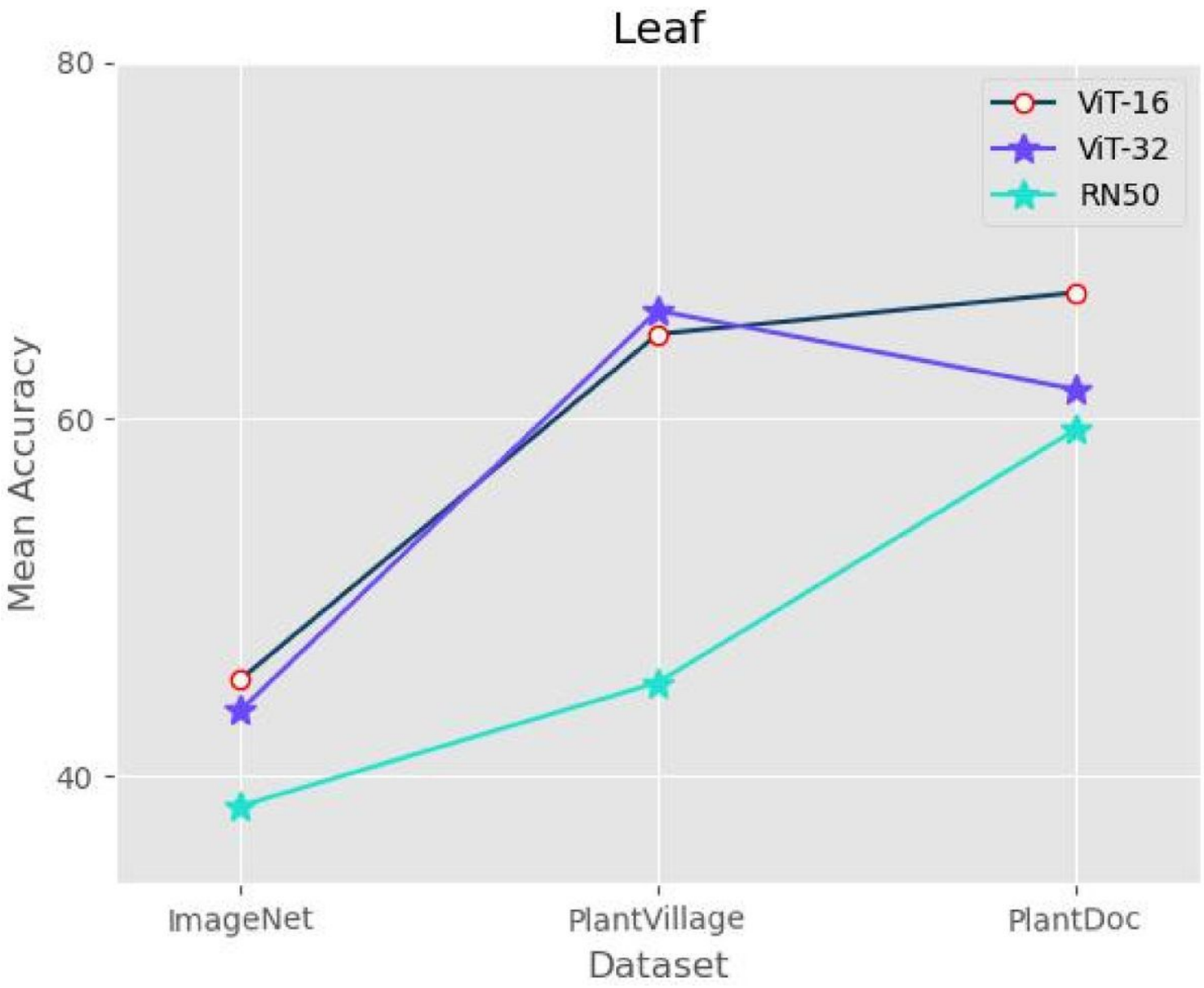

Figure 8

Average accuracy comparison of pre-trained models on different datasets for the classi-fication of apple leaves.

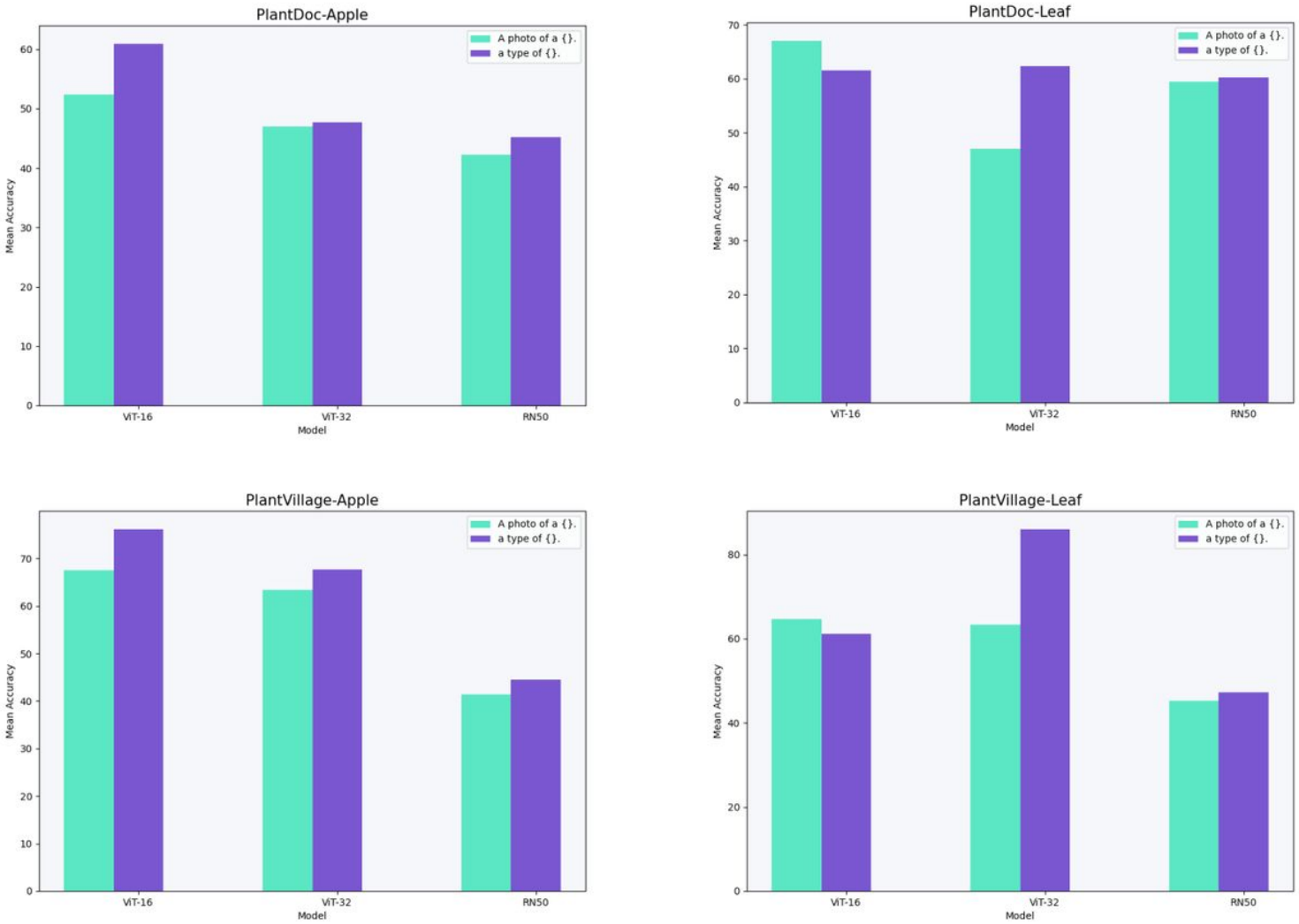

Figure 9

Plot of the average accuracy of the two prompt words on the two datasets (PlantVillage and PlantDoc) compared.

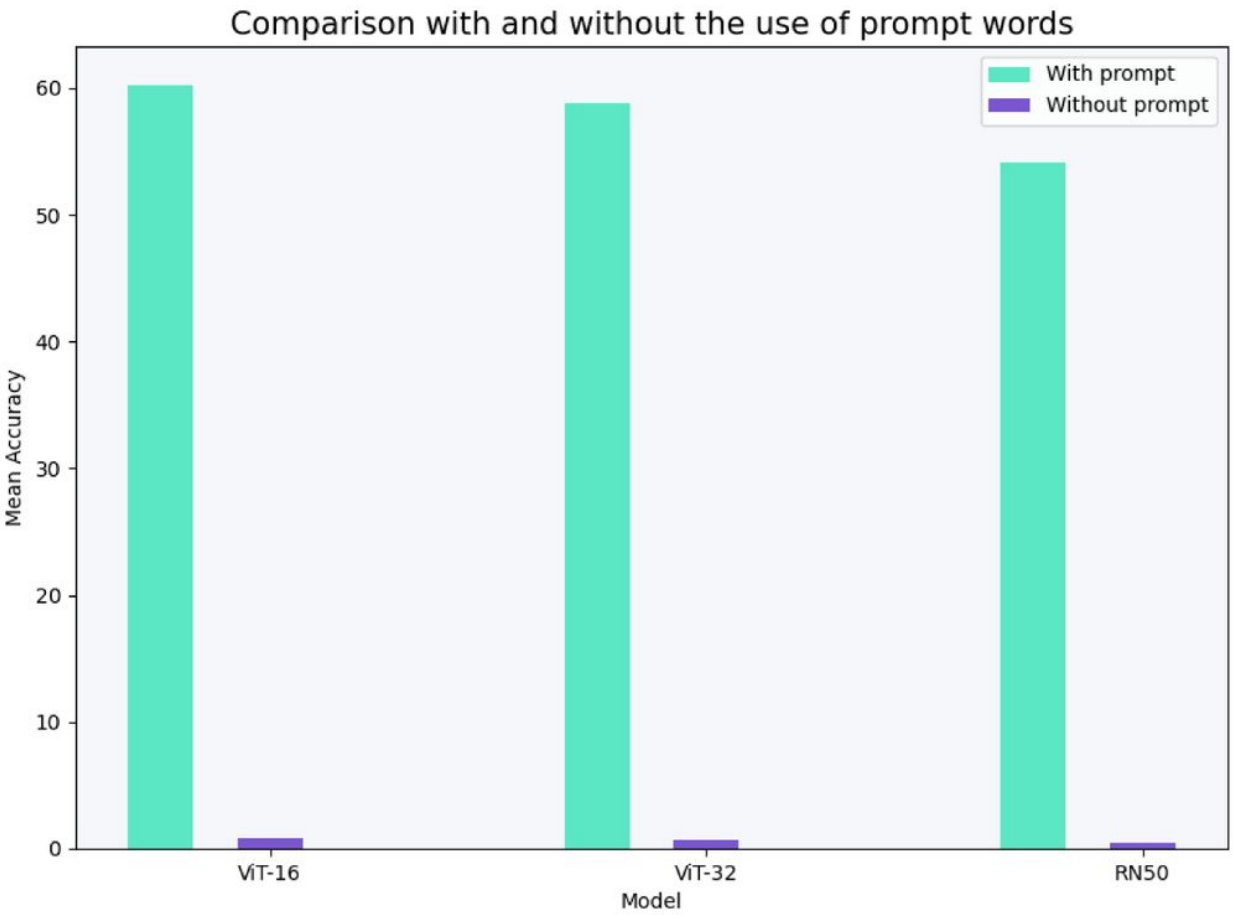

Figure 10

Histogram of average accuracy on the three models with and without the prompt words.

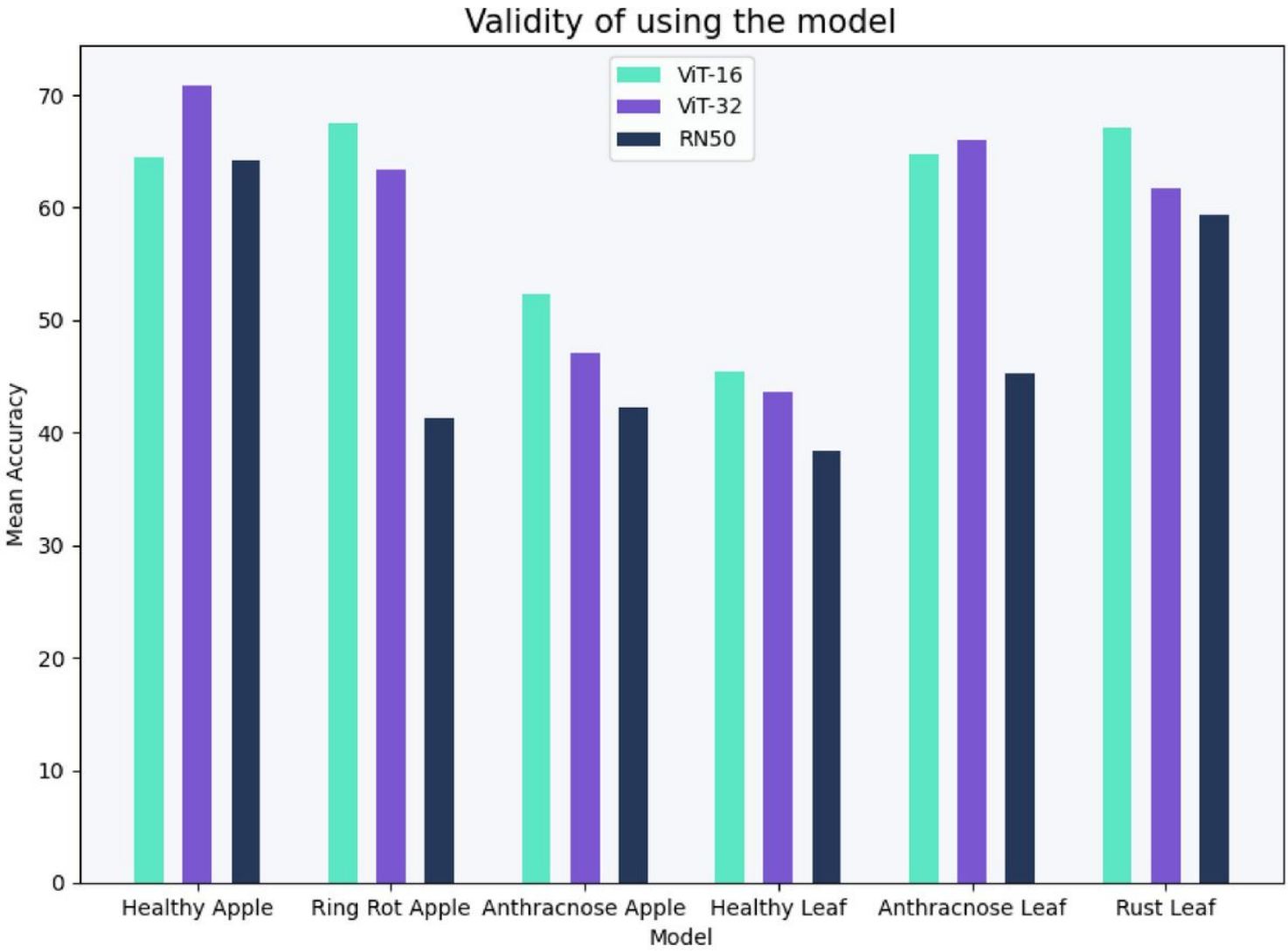

Figure 11

Histogram comparing the average accuracy of each model tested on the Apple dataset.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- TableS1.xlsx
- TableS2.xlsx
- TableS3.xlsx
- TableS4.xlsx