North American Academic Research | Volume 5 | Issue 4 | April 2022 | Monthly Journal by TWASP, USA | Impact Factor: 3.75 (2021)

North American Academic Research

Monthly peer reviewed Journal by **The World Association of Scientists & Professionals**
TWASP, United States

Research Article

Few-Shot Learning: A Step for Cash Crops Disease Classification

Fidel Essuan Nyameke[1*], Shao Bin[1], Raphael K.M Ahiaklo-Kuz[1], Rene Owusu Peprah[1]

[1]School of Information Engineering, Huzhou University, China

OPEN ACCESS

Accepted April 30,2022
Published May 10,2022
 Conflicts of Interest: There are no conflicts to declare.
*__Corresponding Author:__ Fidel Essuan Nyameke
Funding: None

How to cite this article: Fidel Essuan Nyameke, Shao Bin, Raphael K.M Ahiaklo-Kuz, Rene Owusu Peprah (2022). Few-Shot Learning: A Step for Cash Crops Disease Classification. *North American Academic Research, 5(4)*, 283-294. doi: https://doi.org/10.5281/zenodo.6533778

ABSTRACT

Humans' ability to extract information from images is more accessible than machines. The ability of human vision is extraordinary because they have little or no supervision when recognizing objects regardless of the similarity of images. Early studies of visual recognition have shown that machines perform better than humans when there is enough information for prediction and classification. This is less efficient for machines. In this paper, we propose a new way to solve this problem using the provided plant dataset, which will use visualization techniques to solve the problem when the model finds itself in a limited data scenario. Our approach yields more promising results than state-of-the-art models. We used three different types of datasets, including benchmarking Plant Village and Plant Doc. These datasets have controlled, uncontrolled, and downloaded images from the internet. Each dataset is used for our model, resulting in better performance than state-of-the-art results.

Keywords

Few-Shot classification, plant diseases, image recognition, deep learning

Introduction

The exceptional ability of humans to learn with few examples and to generalize classification with highly efficient accuracy has led the artificial intelligence community to open a research direction towards few-shot learning. Assuming taking a kid to the zoo and for the first-time seeing koala once, it will be easier for the kid to identify it again, testing him with a set of animal cards. Computer vision is one of the fastest-growing areas in computing. Numerous approaches have been used to gather information from images—this aid computer scientists in predicting. Prediction is one of the approaches used to make life easier as humans since we are in a time-limited space. The zeal of gathering information from an image started when a summer project was proposed that a camera should be attached to a computer and describe what it saw[1]. This approach is adopted in image recognition, image translation, and image classification[2][3][4]. Humans can identify an object and store its information, but this is different with computers. Computers operate on numbers; therefore, to extract information from an image, there is the need to preprocess the image a represent it with numbers. Images are formed as a unit of many pixels. These pixels are then normalized and represented with numbers[5]. There have been numerous approaches to representing data from images using the neural network approach which has proven to be state-of-the-art[6].In this paper we will use the convolution neural network approach to extract the features. Applications of these computer vision technique has been widely used. Plant diseases has been a problem for some time within the agricultural sector. Cash crops are the one of widely used crops for producing edible product. With cash crops diseases surging high. There is a need to propose a new novel approach which is very simple. Effects of plant diseases has caused more pathogen pollution and using computer vision to minimize the spread will be a lot useful [7]. Using machine learning techniques for recognition, classification, and prediction has enhanced detecting anomalies in plants. This has aided in developing medicines to counter these diseases faster. The interest in this approach has surged higher in using it to predict diseases[8][9][10], analyze sport[11], plant classification[12]. The direct analogy of few-shot learning is the process of a Model being given a large dataset to train on, which is to learn from the function used in distributing the embedding sampled. This step is called learning to learn, learning the function on a large scale and using it to predict unseen data given a small sample from several classes; thus, each class has a few members. This is the problem with few shots or limited data given several classes, n-way and few samples, k-shot as the information to train on and classify the more significant part of the dataset using 20 % data to train and 80% to test. There have been a few approaches, such as Zero-Shot[13] [14], which uses no data to train and strives to classify it's given data; test data can belong to seen or unseen classes. One-shot [15] is the approach used when there's only one sample given per each class of the unseen data to train and classify. We aim to classify disease learning from standardized libraries. A feature extractor is an essential section of a model that hugely improves the classification outcome. Writing a code ground up to extract the essential features of an extensive library and further learning from that to classify would be the best approach but very complex. Instead, we use a straightforward approach. We have come far in this community proposing newer ways of solving this classification problem. There has been some benchmark provided in terms of data in doing this. This solves the initial part of our approach.

Training a function using a large dataset. Fortuitously, most researchers have made these dataset sets available for free. Used by many researchers, ImageNet[16] is the already go-to dataset that can fit most of the newer algorithms and standard libraries developed. Standard libraries developed such as ResNet[17], DenseNet[18] and Inception[19] CNN libraries would be used. All dependencies on Resnet, DenseNet and Inception have proven enough accuracy for feature extraction, therefore our approach would be to deep connect all this CNN features after extraction on the same dataset and use a full connected classifier to classify after we test with our dataset.

Data: The importance of meta-learning and semi-transfer transfer learning is based on the robustness and generalization of the model when fed with new data with few known classes. The reduction in over-fitting determines how versatile the model is for predicting diseases. When training a model for a few shots classification, it is required to be trained on a larger dataset which improves the accuracy based on the diversity of the dataset ImageNet[16]. Based on this reason, we used a public dataset, miniImageNet, containing over a million image datasets for meta-training. Further, we use other public datasets to test the model. The dataset will be used Plant Village Dataset and Plant Doc Dataset. This dataset contains more classes and fewer class members, suitable for a few–shot classification problems. Through this, the model's ability in terms of generalization will improve. A plant dataset is introduced, which is collected as real-world data. This dataset is collected from west Africa, Google images, and Bing imagery using a web-based crawler and classified based on the types of diseases in cash crops. A red-hot disease found in the leaves of the sugarcane rust is also a different type of disease found on the leaf of sugarcane that appears as red dots. Smut usually happens in the stem of the plants, and yellow leaf occurs when the plant lacks enough water. These will be good for a few-shot because of how similar the rust and the red hot are questioning the diversity of the model. We also considered other cash crops.

Feature Extraction: Several methods are proposed to extract features, but CNN has stand-out as the state-of-the-art approach for extracting features over a period. The method proposed for extracting features is encouraged by deep CNN standard libraries that already exist. It makes the process easier by removing the class layer from the top of the CNN and using only the feature extraction for extracting features. Using a pre-trained improves accuracy. Some works have shown a significant increase in accuracy using pre-trained libraries. MPnet trained library has been used in natural language processing[20]. They trained the model on a larger dataset, and further fine-tuned it. The dataset was huge, which will increase time complexity. Due to the accuracy of these standard libraries, it will be able to extract the essential features compared to a network that is built ground up. ResNet, Inception, AlexNet, Densenet all have different approaches for classifying data. This paper intends to use all these deep CNN for extracting features sequentially. This will improve accuracy and decrease overfitting.

Fully Connected Layer: Most state-of-the-art approaches use a least 3 fully connected layers. Recent reviews have shown that the more we increase the fully connected layers, the more we increase overfitting[21]. Investigations have been made for CNN deep learning. A large dataset was used on GoogLeNet, VGG, Resnet using transfer learning[22]. This fully connected layer with building blocks of these state-of-the-art achieved a higher accuracy, with 91.13% of 400,000 documents. The accuracy of this model was improved due to the connected deep CNN layers, which were transferring features through building blocks. The higher the model overfits if a new unknown data is presented. Therefore, this paper proposed an approach to use a fully connected layer with a single hidden layer and a backpropagation technique to update the functions. This will be effective because the feature extractors use a trained approach that will minimize the loss.

Metric Learning: An approach that aids in determining the class of a feature vector is also important in classifying images. Previous researchers use a linear approach for determining a class[23]. Most of these approaches use LSTM, long short-term memory. This approach uses a comparative approach to compare all data with the incoming data to determine the class. Thus, it classifies data by using the closest pair, meaning the element that does not belong to the same class is far away. This raises the question, what about the outliers? This prolongs classification since meta-training is performed in a few shots before the support and the query set are introduced. In this paper, we intend to use the Prototype Network approach, which uses the mean of the embeddings, and nominate a representative that can help determine by using the closest pair approach to determine the class. In this approach, the network represents each class which is a mean because of the grouped elements. This will improve the approach and minimize time complexity.

Optimizer: We are using pre-trained libraries to only extract and not classify, therefore the parameters of the fully connected layer and the hidden layers. This is a non-parametric approach. The loss function used in this paper is the Cross-Entropy, and the optimization is the Adam optimization function. The cross-entropy determines the loss and then updates the fully connected layer.

Classification: Features and labels are matched after every classification problem. In this paper, our main aim is to predict disease in cash crops provided there is limited data, and to do that, we used the few-shot approach. We will use the SoftMax classifier, which uses the cross-entropy function to match the input and the output. It uses a linear dot approach by considering the weight of the feature. This helps in predicting the classes. It also provides the score function that the cross-entropy function will use to update the input of the parameters.

Materials and methods

Problem: In this paper, we solve a limited data problem in deep learning. The traditional technique was to use datasets with each performing their task as train, support, and query, respectively. Both the query and the support have similarities in terms of labeling. We can train the dataset using the support set, but the data to support it is limited. Therefore, with the use of a function that was trained on a large dataset, it can quickly learn faster and is very smart provided the dataset presented to it was limited, which is the support set. Plant disease detection works have shown the classic approach to classifying data feeding the model with more than 60% of the data to be classified with Plant Doc paper explicitly stating that it is difficult for models to perform well if real-world with uncontrolled data settings are presented. The training sets of the model always have their space which depends mainly on augmented data. One of the best approaches would be to use these pre-trained libraries to extract salient information from the data. This would reduce computational loss and increase the use of pre-existing resources.

Dataset: We will use the training dataset to transfer learning. The learning will be observed based on the circle, lines, spaces of the larger dataset, and the second dataset will be used for testing and classifying. This will help us classify the available dataset, such as the Plant Village dataset and the Plant Doc dataset. Table.1 gives the details of the datasets that was used.

dataset	classes			Images		
	total	support set	query set	total	support set	query set
Plant Village	38	5-way-5-shot	5	54303	10860	43443
		10-way-5-shot	10			
Plant Doc	30	5-way-5-shot	5	2569	513	2056
		10-way-5-shot	10			
Cash Crop	10	5-way-5-shot	5	575	115	460
		10-way-5-shot	10			

Table.1 datasets used for testing the few-shot classification model.

***Training Dataset*:** The mini magnet will help us extract some salient information and transfer it to test the datasets we intend to use. We choose to use miniImageNet because of its similarity with the data we are testing with. This miniImageNet is processed from the benchmark ImageNet due to its size. The initial ImageNet is vast, making it challenging to train hence the miniImageNet. It contains 100 classes with 600 images per class selected at random. The total number of the dataset set makes 60000 samples from ImageNet. We use this dataset to train the function for classifying similar diseases in plants.

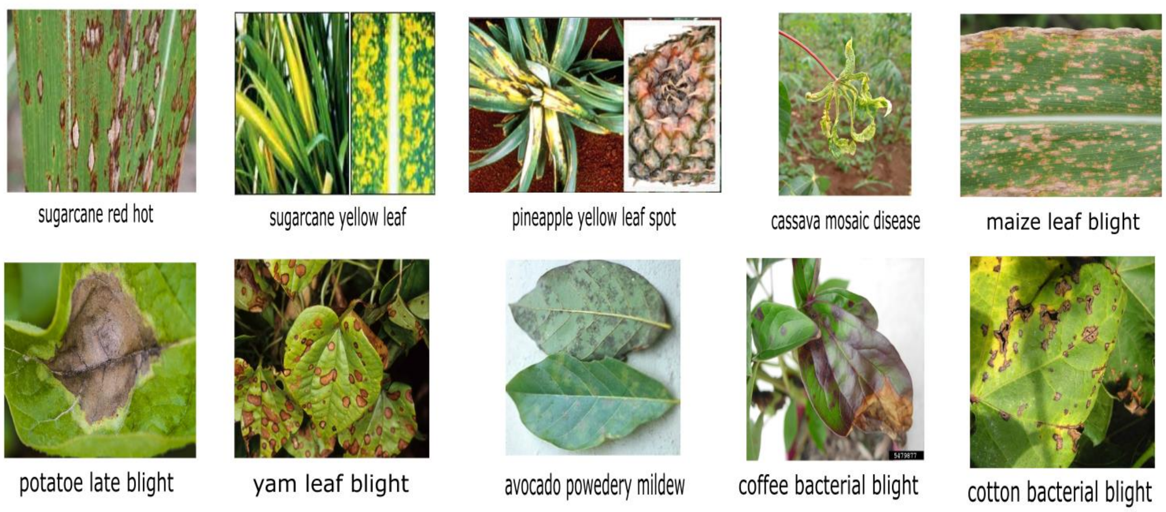

Fig.3 plant disease samples that were downloaded using WebCrawler for testing.

***Testing Dataset*:** We use datasets for testing. These datasets include healthy plants, leaves, and affected plants, leaves. We are trying to propose a few-shot learning approach that will make the technique more agile to classify plant diseases even when the data settings are not controlled. Plant Doc dataset is a set of data that contains over 2,598 samples for 13 species of plants and 27 classes. This includes 17 which are infected plants and 10 which are healthy ones. This dataset provides a benchmark for plant disease classification. The Plant Village dataset is another benchmark provided publicly for researchers. This dataset contains healthy leaf and unhealthy, which has been categorized into 38. This is made up of 87000 species and diseases. The plant disease ranges from apple, orange, peach, potatoes, and tomatoes, and finally, we use datasets that include cash crops.

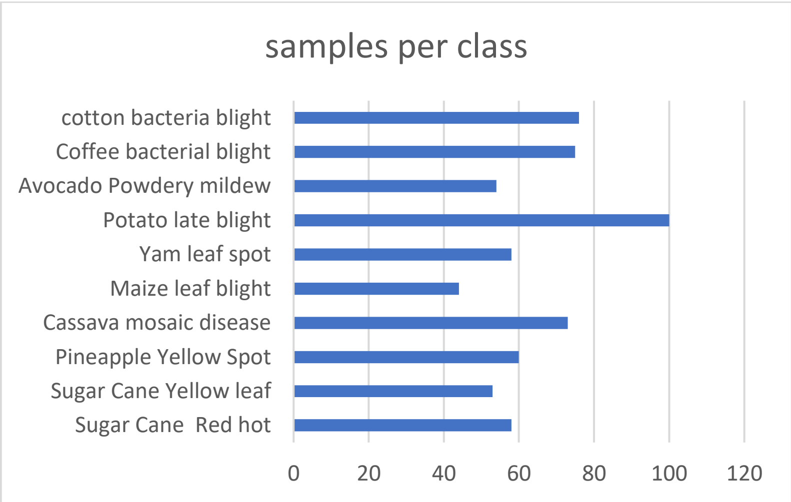

Fig.1 Downloaded images using a WebCrawler[1] tool for Cash crop Dataset.

Model

This section will dissect our model and explain the procedure we would like to use for few-shot learning in plant disease classification. The beginning analyzes the problem of limited data and why this scope will aid in bridging that gap when dealing with real-world data. Our model has three parts to initiate the concept of meta-learning. Meta-learning would be discussed as improving function agility based on a large dataset. This makes the model work faster and more accurately. The first part of our model uses pre-trained libraries, which are Resnet and Densenet. With the aid of the PyTorch[24] framework, we can use the technique to fetch functions that have been trained with ImageNet dataset. We will use these features sequentially; this will aid the model in reducing loss at the start of the procedure before we get to the final stages of the classification model. We further our implementation with the use of cross-entropy loss to minimize losses. With backpropagation, the model can update the loss of the model. Adam function will be used since it's agile and has proven to work well with most models. Finally, we will fuse two loss functions which were hybridized

[1] https://github.com/sczhengyabin/Image-Downloader

functions used by [25]. The figure below shows the first part where we save the extracted features, and the second shows how we continue to use the extracted features for classification.

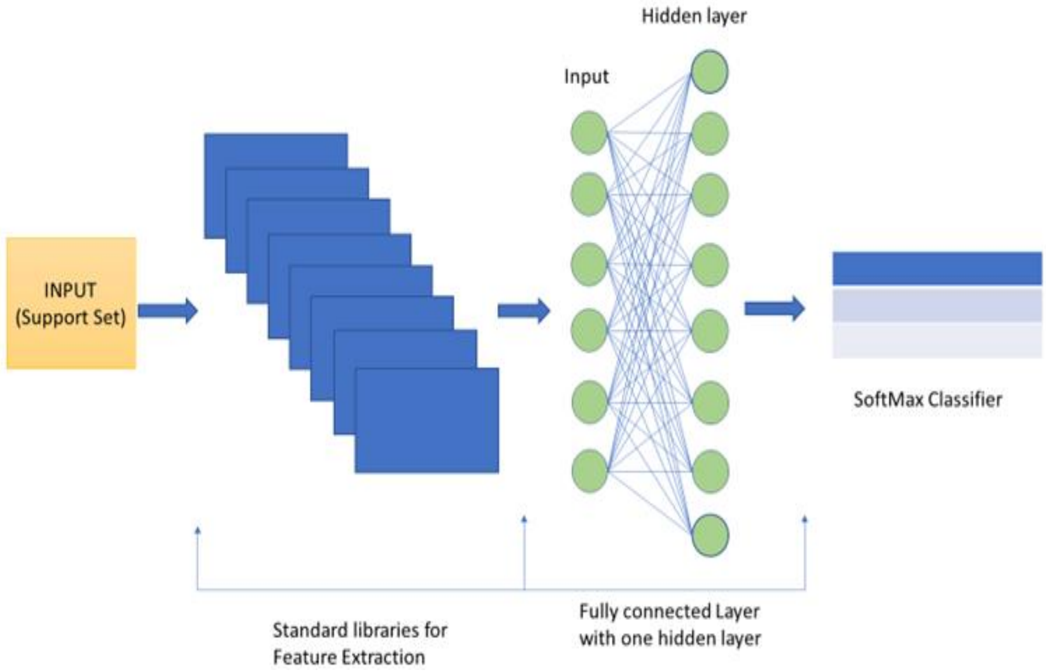

Fig.2 First part of the model shows how the pretrained libraries were used sequentially to extract the features.

Feature Extraction: The pretrained libraries use a convolutional neural network for image numerals computations. The libraries architectures are the same as the traditional CNN, with only the layer which forms the classification layer being blocked. The parameters to update the model for prediction are also not updated. Images are passed through these libraries in batches. These essential features will be saved to be used sequentially. Reset and Densenet libraries are the only pretrained libraries that will be used in this work.

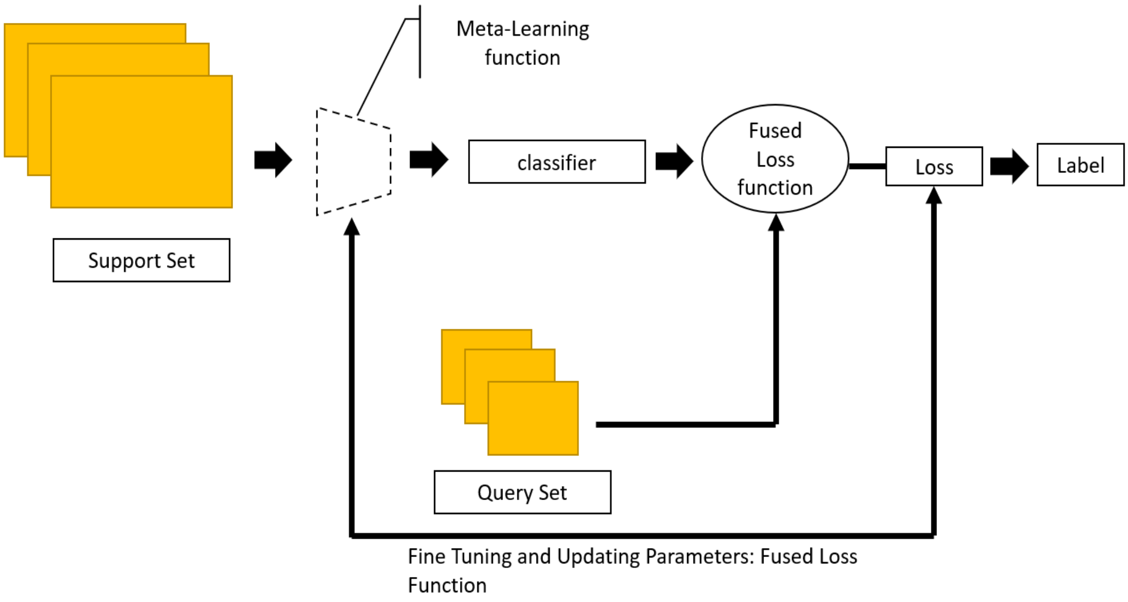

Fig.3 classification network of the model using Fused loss function.

***Loss Function*:** The loss function is the difference that is observed between the actual happenings and the predicted ones. Let's represent a tuple of x, y as the data unit. These tuples contain the chrematistics of the data unit as well. The model uses these to compare them with the predicted value showing the error margin.

Optimization: This forms part of the network that uses a mechanism to update the loss calculated by the loss function. In this paper, the pretrained library, which is for transfer learning, saves time, making the technique more efficient. We intend to use the general optimizer used by a lot of models—Adam optimizer and set a batch size of 128. Therefore, for every 128-batch size, the parameters will be updated.

Parameter optimization and the learning rate: Like the model progress through the layers, the weights are divided and reduced with a lot. This could lead to an error because of the size of the features that needed to be learned. Overfitting could also occur. Hence we set our rate to 0.09; hence it will learn a new pattern after every 0.09 step. After the division and the update are learned, it will update the network. This goes on until the image batches are done. With hyperparameters set, if the accuracy of the model is improved, it will stop its procedure

Few-shot learning has two benchmarks dataset for meta-training. The first is the miniImageNet[16]. In this paper, we will be using the miniImageNet for meta-training. The miniImageNet will help us to extract some salient information and transfer it for testing the datasets we intend to use. We choose to use miniImageNet because of its similarity with the data we are testing with. This miniImageNet is processed from the benchmark ImageNet [16] due to the size. The initial ImageNet is vast, making it challenging to train hence the miniImageNet. It contains 100 classes with 600 images per class that were selected at random. The total number of the dataset set makes 60000 samples from ImageNet. We will use this dataset to train the function for classifying similar diseases in plants. This was downloaded from which has been already preprocessed to 224 x 224. The dataset contains images that have been gather from the internet to be used for research purposes. Data augmentations are already done before used. This is because of the framework the is proposed by PyTorch to be used for transfer learning. Making easier for deep CNNs to transfer knowledge from learning the circles, shapes of a huge different set of data. miniImageNet makes the workflow easy and improves the performance of the function to classify plant diseases on real-world data. We use 2 forms of datasets for testing. These datasets include healthy plants, leaves, and affected plants, leaves. We are trying to propose a few-shot learning approach which will make the technique more agile to classify plant diseases even when the data settings are not controlled. Plant Doc dataset is a set of data which contains over 2,598 samples for 13 species of plants and 27 classes. This includes 17 which are infected plant and 10 which are healthy. This dataset provides a benchmark for plant disease classification. The Plant Village dataset is also another benchmark that was provided publicly for researchers. This dataset contains healthy leaf and unhealthy which has been categorized into 38. This is made up of 87000 species and diseases. The plant disease ranges from apple, orange, peach, potatoes, and tomatoes and finally we use datasets which includes cash crops. This dataset will not be controlled (not augmented).

Testing: Datasets are randomly selected and grouped into different categories. The first instance is each dataset will be grouped into 5 classes, and each class will contain 5 data samples. The second instance will be to use 10 classes and 5 data samples. The only class in which data samples are not randomly selected is the one that we downloaded from the internet. These datasets include 10 different classes of different diseases that affect cash crops. In the previous chapter, we shared a table that includes the amount of data was captured by the image search engines Google[2] and Bing[3]. We used a web scraper tool[4]. The dataset was cleaned, removing unwanted images. The Plant Village dataset[5] is essential in this work because it provides us with both healthy and unhealthy plant species. It also provides us with augmented datasets that are not controlled. In-Plant dataset, there are 38 classes of plants; this has 17 healthy species and the rest unhealthy. Most of the images are leaves of crops that are either affected by a disease or not. The total dataset is 50,000 images. We randomly selected 5 samples for 5 classes and trained them for 10 trials. We also randomly selected 5 samples from 10 classes. We chose the number of trials due to computation cost. Plant Doc dataset[6] was that was proposed as means to provide a dataset for plant diseases. This provided us also with diversities based on the data we are working around and the aim of this paper. It has been provided because of an effort to crowdsource information on plant diseases. The Plant doc dataset included 17 classes of 13 species that have been infected with diseases. The class of this dataset is based on the diseases, not the species. In this paper, we randomly selected 5 samples from each class which is the first instance, and the second instance follows. The number of trials was 5. We didn't augment the datasets because it was already cleaned, and the aim of this paper is to determine how a good model can classify the uncontrolled dataset.

Analysis: During the initial steps, we analyzed that the ProtoNet performed better than the MAML because considering metric distances in limited scenarios were efficient than only fine-tuning the model. For both instances, the model fused with hybridize loss function achieved 85% and 87% for uncontrolled dataset settings, although the performance was less effective. The state-of-the-art models performed well, but without the pre-trained libraries, the initial stages of the networks will experience some losses, which will affect the performance of the model. We discussed the scenarios we created to implement randomly picking samples a using them for testing and analysis. We also provided the average performance table of our model.

	Dataset	MAML	ProtoNet	Model	Model + LF function

[2] https://www.google.cn/imghp?hl=en&ogbl
[3] https://cn.bing.com/visualsearch
[4] https://github.com/sczhengyabin/Image-Downloader
[5] https://data.mendeley.com/datasets/tywbtsjrjv/1
[6] https://github.com/pratikkayal/PlantDoc-Dataset

5-way 5-shot	Plant Village	50.70%	52.80%	73.30%	92.30%
	Plant Doc	51.10%	54.50%	74.30%	93.40%
	Cash Crop	49.50%	50.90%	69.90%	85.30%
10-way 5-shot	Plant Village	50.92%	51.20%	78.40%	92.50%
	Plant Doc	55.30%	56.40%	79.40%	93.50%
	Cash Crop	50.20%	52.10%	70.90%	87.50%

Table 2. The test results of our model use the benchmark datasets and our generated dataset.

Conclusion

Humans have the exceptional ability to detect and classify data. This is one of the skills that, to date, human has over the computer. Over the decade, the computer has helped us to improve our lives and become more efficient in what we do. Plant disease is increasing, as has been discussed in this paper. Computer vision has gradually made an impact in this area of discipline that solves classification problems. In this paper, we analyzed a technique that addresses the problem of limited data and uncontrolled datasets. We observed that although the model performed less than controlled datasets, it showed a good percentage of 85.30% and 87% when there were more classes. Showing that the more support dataset we have, the better the classification. In conclusion, we showed the importance of fused loss function and the importance of pretrained libraries. These are the essential parts of model tackling few-shot learning. Firstly, we suggested more work needs to be done to provide datasets that become a benchmark in this disciplinary field. Also, most of the dataset used for meta-training was massive costing computation. We achieved a poor performance for the cash crop datasets because we couldn't annotate them well.

References

[1] S. Papert, “The summer vision project.” pp. 1–6, Jul. 01, 1966, Accessed: Nov. 09, 2021. [Online]. Available: https://dspace.mit.edu/handle/1721.1/6125.

[2] S. Qiao, C. Liu, W. Shen, and A. Yuille, “Few-Shot Image Recognition by Predicting Parameters from Activations,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 7229–7238, 2018, doi: 10.1109/CVPR.2018.00755.

[3] M. Y. Liu *et al.*, “Few-shot unsupervised image-to-image translation,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 10550–10559, 2019, doi: 10.1109/ICCV.2019.01065.

[4] P. Bateni, J. Barber, J.-W. van de Meent, and F. Wood, “Enhancing Few-Shot Image Classification with Unlabelled Examples,” 2020, [Online]. Available: http://arxiv.org/abs/2006.12245.

[5] S. J. H. Ma, “Image Normalization, a Basic Requirement for Computer-based Automatic Diagnostic Applications.,” *Image (Rochester, N.Y.)*, vol. 44, no. 0, pp. 0–26, 2001, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.8819&rep=rep1&type=pdf.

[6] P. C. P. Júnior, A. Monteiro, R. da L. Ribeiro, A. C. Sobieranski, and A. von Wangenheim, “Comparison of classical computer vision vs. Convolutional neural networks for weed mapping in aerial images,” *Rev. Inform. Teor. e Apl.*, vol. 27, no. 4, pp. 20–33, 2020, doi: 10.22456/2175-2745.97835.

[7] P. K. Anderson, A. A. Cunningham, N. G. Patel, F. J. Morales, P. R. Epstein, and P. Daszak, “Emerging infectious diseases of plants: Pathogen pollution, climate change and agrotechnology drivers,” *Trends Ecol. Evol.*, vol. 19, no. 10, pp. 535–544, 2004, doi: 10.1016/j.tree.2004.07.021.

[8] A. Dutta, T. Batabyal, M. Basu, and S. T. Acton, “An efficient convolutional neural network for coronary heart disease prediction,” *Expert Syst. Appl.*, vol. 159, 2020, doi: 10.1016/j.eswa.2020.113408.

[9] E. Hosseini-Asl, R. Keynton, and A. El-Baz, “Alzheimer’s disease diagnostics by adaptation of 3D convolutional network,” in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, vol. 2016-Augus, no. Icip, pp. 126–130, doi: 10.1109/ICIP.2016.7532332.

[10] S. Baek *et al.*, “Deep segmentation networks predict survival of non-small cell lung cancer,” *Sci. Rep.*, vol. 9, no. 1, pp. 15–17, 2019, doi: 10.1038/s41598-019-53461-2.

[11] S. Giancola and B. Ghanem, “Temporally-Aware Feature Pooling for Action Spotting in Soccer Broadcasts,” pp. 4485–4494, 2021, doi: 10.1109/cvprw53098.2021.00506.

[12] M. Brahimi, S. Mahmoudi, K. Boukhalfa, and A. Moussaoui, “Deep interpretable architecture for plant diseases classification,” *Signal Process. - Algorithms, Archit. Arrange. Appl. Conf. Proceedings, SPA*, vol. 2019-Septe, pp. 111–116, 2019, doi: 10.23919/SPA.2019.8936759.

[13] S. Narayan, A. Gupta, F. S. Khan, C. G. M. Snoek, and L. Shao, “Latent Embedding Feedback and Discriminative Features for Zero-Shot Classification,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12367 LNCS, pp. 479–495, 2020, doi: 10.1007/978-3-030-58542-6_29.

[14] G. Dinu, A. Lazaridou, and M. Baroni, “Improving zero-shot learning by mitigating the hubness problem,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Work. Track Proc.*, pp. 1–10, 2015.

[15] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese Neural Networks for One-shot Image Recognition.”

[16] L. Fei-Fei, J. Deng, and K. Li, “ImageNet: Constructing a large-scale image database,” *J. Vis.*, vol. 9, no. 8, pp. 1037–1037, 2010, doi: 10.1167/9.8.1037.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[19] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07-12-June, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[20] K. Song, X. Tan, T. Qin, J. Lu, and T. Y. Liu, "MPNet: Masked and permuted pre-training for language understanding," in *Advances in Neural Information Processing Systems*, 2020, vol. 2020-Decem, no. NeurIPS, pp. 1–14.

[21] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, doi: 10.1162/NECO_a_00990.

[22] M. Z. Afzal, A. Kolsch, S. Ahmed, and M. Liwicki, "Cutting the Error by Half: Investigation of Very Deep CNN and Advanced Training Strategies for Document Image Classification," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2017, vol. 1, pp. 883–888, doi: 10.1109/ICDAR.2017.149.

[23] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017, pp. 1–11.

[24] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, vol. 32, no. NeurIPS.

[25] G. B.V. and U. D. G., "Identifying and classifying plant disease using resilient LF-CNN," *Ecol. Inform.*, vol. 63, no. February, p. 101283, 2021, doi: 10.1016/j.ecoinf.2021.101283.