AN EFFICIENT EDGE-BASED PLANT DISEASE DETECTION MODEL USES AN ENRICHED DATASET AND DEEP CONVOLUTIONAL NEURAL NETWORK

Anh-Thu Pham, Van-Nhan Nguyen, Tuan-Anh Pham, Thanh-Tra Nguyen
Telecommunications Faculty No1, Posts and Telecomms Ins of Technology Hanoi, Vietnam

Abstract–Crops and yields are significantly harmed by plant diseases, one of agriculture's most significant problems. Researchers have recently investigated using artificial intelligence (AI) to detect and effectively manage disease early on to address this issue. This research focuses on developing a method to optimize the DCNN (Deep Convolutional Neural Network) classification model for plant diseases. We enriched the data by incorporating data from two public datasets, PlantVillage Dataset (PVD) and CroppedPlant Dataset (CPD), and we trained the model using two-step transfer learning. The experimental results demonstrate that the model's accuracy is 82%, more significant than previous studies. Notably, achieving this result with fewer parameters while maintaining adequate performance compared to previous research demonstrates the model's efficient use of limited computing resources. Hence, the proposed model is deployable on edge devices to optimize availability and efficiency in real-world environments and contribute to deploying new edge computing and agriculture services.

Keywords– Leaf Diseases, Data Augmentation, Transfer Learning, Edge Computing

I. INTRODUCTION

The impact of plant leaf diseases on crop quality and production poses several challenges for agriculture today. These diseases can cause significant crop damage, reducing yields and quality while increasing production and management costs. Artificial intelligence, specifically image processing and deep learning techniques, can be used to detect and identify leaf diseases, which is a promising and practical solution [1] [2] [3].

Due to its convenience and efficiency, there is a pressing need for an automatic and accurate method to detect and identify plant leaf diseases [4]. Deep convolutional neural networks (DCNNs) have emerged as one of the most prevalent and effective deep learning techniques in image processing [5]. DCNNs can learn and apply sophisticated image features to tasks such as segmentation, classification, recognition, and object detection. In addition, deploying AI models on edge computing devices is becoming an emerging approach in 5G and beyond for real-time and low-latency applications [6].

Mixing datasets is an approach for increasing the diversity of datasets and enhancing model generalization [7] [8]. However, it may also occur in overfitting and increased computational costs [9] [10]. To overcome these challenges, we combine the PlantVillage Dataset (PVD) dataset [11] with the Cropped PlantDoc dataset [12] to enhance our diagnostic accuracy. In recent years, transfer learning techniques, particularly the two-step transfer learning approach, have demonstrated their effectiveness in enhancing the performance of deep learning models [13] [14]. This study uses a two-step transfer learning technique to reduce the computational cost before putting the data into our lightweight DCNN model (MobilenetV3large). Our experimental results indicate that our proposed method obtains better performance metrics than other state-of-the-art studies. This paper is structured as follows. The following section presents related work. Section 3 summarizes the characteristics of the two datasets utilized in the model and the system's overall architecture for image-based disease diagnosis. Section 4 provides our experimental results that compare the performance metrics with other studies. The last section provides our conclusions and future research directions.

II. RELATED WORK

Recent developments have been made in classifying leaf images using AI models for plant disease identification. Deep learning models for the PlantVillage Dataset (PVD) dataset [11] have obtained extremely high accuracy [15]. The constrained image capture conditions, which are difficult to acquire in real life, are a limitation of these models. Therefore, the Cropped PlantDoc (CPD) dataset [12] containing various real-life images has a more significant practical application. However, the efficiency of AI methods on this dataset still needs to be improved. The authors of [16] proposed kEffNet-B0, an enhanced deep CNN model based on EfficientNet-B0 that achieved 64.39 percent accuracy. Another study [17] by the author [16] using kEffNetB0- 32ch improved the accuracy better [16] by 65.74%. In [12], the authors used InceptionResNet-V2 with an accuracy of 70.53 percent. With CPD dataset preprocessing, the authors [18] obtained 77% classification accuracy. The DCNN model in our

Contact author: Anh-Thu Pham,
Email: thupa@ptit.edu.vn
Manuscript received: 10/2023, revised: 11/2023, accepted: 12/2023.

previous work is quantized to be more suitable for edge devices. Our model work utilizing a lightweight DCNN (MobilenetV3Large) achieved approximately 78% accuracy for multi-class CPD classification [19]. Despite the use of some lightweight DCNN models that can be deployed on edge devices, the efficacy of these models on real-world image datasets still presents challenges, according to these studies. Therefore, the main contribution of this study is a new framework that combines dataset mixing, transfer learning, and DCNN techniques to increase the accuracy of plant disease diagnosis in natural image datasets.

III. ENRICHING THE DATASETS

This section will present two popular data sets used in the agricultural field. We enrich the datasets by mixing natural data into the ideal data, called data preprocessing.

A. Data sets

Detecting plant diseases is an essential task in agriculture but is still difficult because of the impacts of environmental conditions. Computer vision technologies have proven to be an effective tool for disease detection. However, the different datasets used by different DCNN models may produce different results. To demonstrate the effectiveness of DCNN models, the models mentioned above often use laboratory datasets (PlantVillage Dataset) and real-life datasets (Croped- PlantDoc). Each data set has its characteristics as well as its advantages and disadvantages. Below, we will present the overview of these two data sets.

PlantVillage Dataset [11]: Data sets used in agriculture often require an extensive, validated database of photos of healthy and damaged plants to develop accurate image classifiers for diagnostic applications of plant diseases. Such a dataset did not exist until recently, and even smaller datasets are not publicly accessible. To address this problem, the PlantVillage Dataset project initiated the collection of thousands of images of healthy and diseased plants that were freely accessible to the public. All images in the PlantVillage Dataset database were taken at experimental research stations and laboratories, with various brightness, environment, and other user-specified settings. Finally, the end devices (smartphone users) will take pictures in various” random” conditions. Figure 1 shows an example of potato images in the Planvillage dataset.

More than 50,000 images in the Planvillage dataset are currently hosted on the website www.PlantVillage Dataset.org, and this dataset can be accessed through US universities (Penn State, Florida State, Cornell, and others). The dataset contains 54,303 images of healthy and unhealthy leaves, classified into 38 categories based on species and disease. Plants such as Apples, Blueberries, Cherries, Corn, Grapes, Oranges, Peach, Bell Peppers, Potatoes, Raspberries, Soybeans, Pumpkins, Strawberries, and Tomatoes are all included in that dataset. In addition, illustrations of 17 fungal diseases, four bacterial diseases, two mold diseases (oomycetes), two viral diseases, and one tick-borne disease are also shown in that dataset. There are images of healthy leaves on 12 disease-free plant species, and the total number of classes in the dataset is 38.

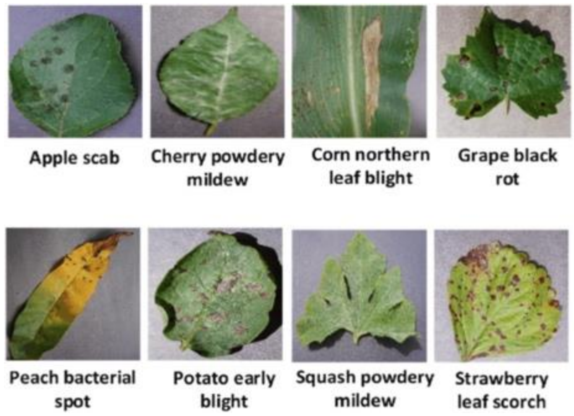

Fig. 1. Some of the plant diseases from the PlantVillage Dataset

The Cropped-PlantDoc dataset [12]: Singh and his collaborators created the Cropped-PlantDoc dataset containing 13 plant species and 27 classes. Similar to the Plant Village dataset, the original PlantDoc dataset includes an image of each leaf. However, those images also show the complex background and the area covered by the different target leaves, which makes classification much more difficult than the PlantVillage Dataset images. The authors manually crop the image regions containing the target leaves to address this shortcoming. This produces conveniently framed leaves while greatly increasing the number of samples (approximately 9K) as several samples from each original PlantDoc image can be extracted (approximately 2.6K). Figure 2 presents some examples of leaf images in the PlantDoc Dataset.

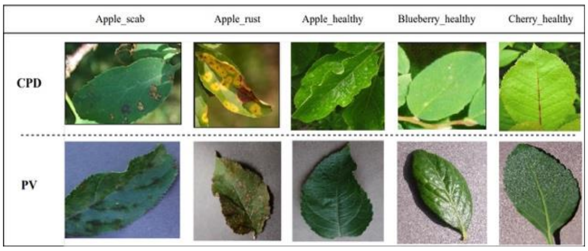

Fig. 2. Example images of CPD and PVD

B. Data preprocessing

As mentioned at the beginning, we combine two datasets in this paper: laboratory dataset (PVD) and naturally collected dataset (CPD). However, there is a problem with the number of classes in the two datasets; that is, the PVD dataset has 38 classes, but the CPD dataset has only 27 classes. Since the 27 classes in the CPD dataset are all included in the PVD dataset, and this study aims to make the model deployable in practice, we will test the model performance on the combined CPD dataset and 27 of the 38 classes of the PVD dataset. After combining, the new dataset will have 27 classes similar to the CPD dataset.

This study combines two datasets, including PVD and CPD, as illustrated in Figure 3. First, the PVD dataset is randomly divided into two subsets: PVD train and PVD val. This division is performed in three different cases with the division ratios of 80:20, 70:30, and 50:50, respectively. Similarly, the CPD dataset is randomly divided once to form three subsets: CPD train, CPD val, and CPD test, with a ratio of 65:15:20.

The data merging process is applied only during the training and validation phases. When conducting the final model testing, we focus on the more complex dataset, i.e., the CPD dataset. We perform the matching in pairs: CPD train is combined with the PVD train, CPD val is combined with PVD val, and the CPD test is kept intact. The result of this combination creates a dataset called 'Combine data,' consisting of three cases corresponding to three PVD split ratios: Combine data 1, Combine data 2, and Combine data 3. Each of these combined data consists of three subsets of data, which are used for training, validation, and testing, respectively. By comparing and analyzing the model's results on each of the three PVD dataset division ratios, we would like to determine our model's most optimal partitioning case.

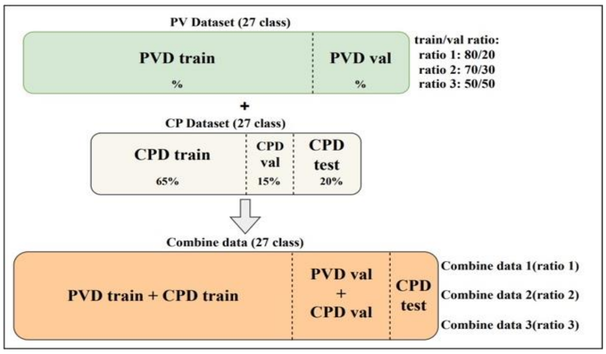

Fig. 3. The combined dataset of the CPD and PVD

In addition, data augmentation is also applied before feeding the training model. The augmentation techniques used include Randomly Flipping the images horizontally, Randomly rotating images with a maximum angle of 30 degrees, Randomly zooming the images with a maximum factor of 30%, and Changing the contrast of the images with a maximum factor of 30%.

IV. THE PROPOSED MODEL AND RESULTS

A. The proposed DCNN model

In this paper, we use the MobilenetV3Large model in the previous study [19], and there are some changes in model training. The MobilenetV3 model has two variants, including MobileNetV3-Small and MobilenetV3-Large. These two variants have the same architecture and only differ in the number of parameters. Transfer learning in image processing has improved with time, identifying efficient kernels/filters or models. These well-known pre-trained models use the knowledge gained from training on thousands of objects in the ImageNet dataset, the largest repository of its kind. Integrating a pre-trained model with ImageNet into a framework uses kernels that match leaf properties as a starting point. This method uses verified and correct knowledge to fine-tune kernels that may not match plant ingredients.

Our pre-training uses a MobileNetV3Large extractor to extract features from the input Imagenet dataset (Figure 4). The initial weight of the extractor is learned from the dataset. Then, we add a 'global average pooling2d' layer to reduce the output size of the network and, at the same time, retain the image feature after extraction. In addition, the 'dropout' layer is added after the 'global average pooling2d' layer to prevent over-matching. Finally, MobileNetV3's extractor is retrained on the combined dataset to update the original weight of the extractor.

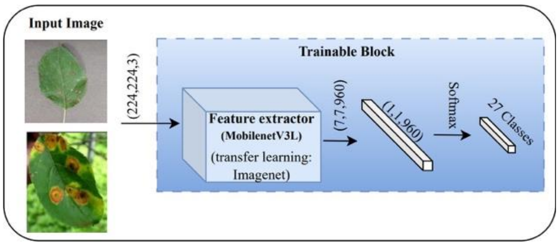

Fig. 4. Pre-training feature extractor

At the end of pre-training, the model will be finetuned. The output of the global average pooling2d class will be extracted. This helps us obtain the extracted image's feature after passing through the extractor for further classification processing. After the extractor is pre-trained, the DNN classifier is placed at the extractor output. Precisely, 4 Dense classes with the number of nodes 512, 512, 128, 128, and 27 correspond to 27 classes in the dataset, using the activation function "Relu." They are placed after the 'dropout' layer in the extractor, as shown in Figure 5. The final model has 3.8 million parameters, and finetuning is done with this new model. During the finetuning process, we finetuned the model on the dataset with only CPD data by only updating the classifier's parameters. The previous layers in the feature extractor will be frozen. The final model validation and testing processes will also be performed on the CPD dataset.

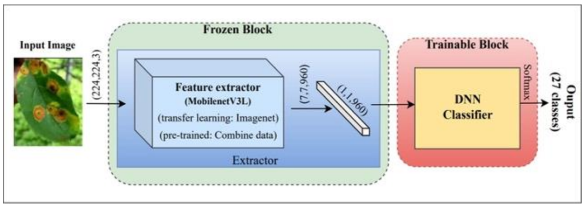

Fig. 5. The proposed training model with pre-trained extractor and DNN classifier

B. Training results

Our experiment is implemented in Python 3.7 with the TensorFlow framework and Keras library for Deep Learning tasks. Experiments were performed on a computer featuring an Intel ® Core i9 10900K, a Nvidia® RTX A4000 GPU, and 48 GB RAM. We train to achieve the best model with the following training results. After

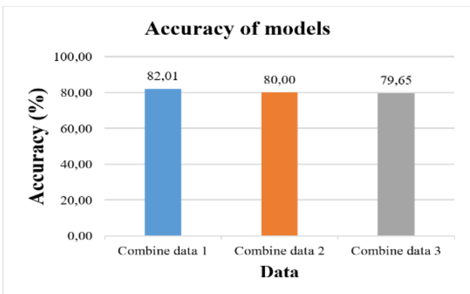

Fig. 6. Accuracy of the model on three sets of Combine Data

training, the model results on 3 cases of the Combine data are presented in Figure 6.

As shown in Figure 6, the model could achieve the highest accuracy with the dataset of Combine data 1, which corresponds to dividing the data in the PVD set with a ratio of 80:20. Since the model achieves the best accuracy on Combine data 1, in the next section, we will select the results of the model on this dataset to compare with the results of the other authors. Figure 7 shows the results of the extractor training and validation.

The accuracy graph shows that the training process took place smoothly in the first ten epochs, so the model learned from about 70% up to 96%. After those ten epochs, the training accuracy increased slightly and stabilized at 98%. Likewise, validation accuracy after ten epochs also starts to stabilize around 96%.

Similar to the Accuracy graph, the Loss graph in 8 during model training also quickly reached the optimal level after ten epochs. After the 10th epoch, the loss of both the training and validation processes decreased slightly and stabilized. Moreover, after 50 epochs, both the Accuracy and Loss of the model have reached the learning threshold.

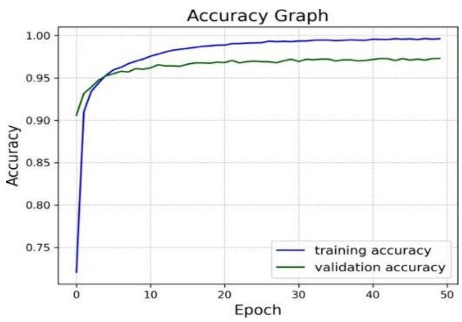

Fig. 7. Accuracy of training and validation

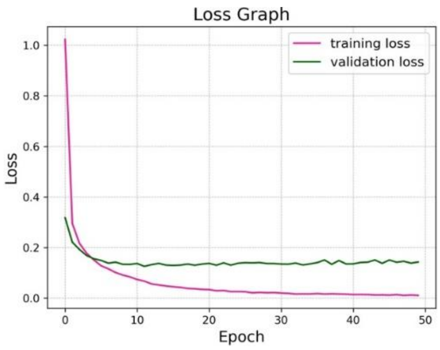

Fig. 8. Loss of training and validation

C. Parameters and comparison results

In this section, we compare the previously studied DCNN models in terms of parameters and accuracy. The specific results related to the model parameters and the model accuracy given by the authors mentioned in part II are shown in Table I.

Table 1: A comparison of the dcnn model in terms of parameter and accuracy

Model	Year	Parameters	Accuracy
EfficientNet [16]	2021	664K	64.39%
kEffNet-B0 32ch [17]	2022	1.08M	65.74%
InceptionResNetV2 [12]	2020	“	70.53%
Color-Aware Two-Branch [18]	2022	5M	76.91%
MobilenetV3-Large [19]	2022	5M	77.71%

The results in Table I have shown that the model proposed by the authors in [16] has the lowest number of parameters of 664K, leading to the model's accuracy only stopping at 64.39%. In addition to optimizing the number of parameters to deploy on edge devices, accuracy must also be ensured at a reasonable level. Our previous study [19] achieved an accuracy result of 77.71%, higher than that of previous authors. Specifically, the accuracy result given by the authors in [18] is 76.91% when having the same number of model parameters. In this study, we continue to improve the model's accuracy and number of parameters to compare it with our previous study [19].

Table 2: Comparison of the proposed model with the previous model

Model	Parameters	Accuracy	F1-Score
MobilenetV3large [19]	5M	77.71%	0.7723
Proposed Model	3.8M	82.01%	0.8194

The comparison results between the proposed model in this paper and the proposed model in the previous study are described in Table II. The model proposed in this article has achieved better parameters, accuracy, and F1 score than the previous model. Specifically, the accuracy of the proposed model is 5% better than previous research [19]. Next, the F1- Score of this model is also higher than that of the previous study. In addition, the number of parameters of the proposed model is also focused on when developing the model on edge devices. The model in this study achieves higher accuracy and requires fewer parameters than in previous studies. The fact that the proposed model achieves a testing accuracy of about 82% while training accuracy reaches 98% might be worth considering. The difference in the training dataset and testing dataset can explain the cause. We use a combined dataset during training, including the laboratory and real-life datasets. Training datasets from the laboratory can be simple and give better results, so the accuracy in this process is usually higher. However, it is easy to see that the real-life dataset in the combined dataset is less than the data from the laboratory. This can lead to high accuracy during training that does not accurately reflect the model's ability for real-life data. When we conduct model testing, we only use real-life datasets, which requires the model to deal with real-life images with more complex backgrounds than those in the lab. This can significantly reduce the model's accuracy compared to the training process.

V. CONCLUSION

This paper presents a methodology for enhancing input data and a transfer learning approach for a DCNN model dedicated to plant leaf disease classification. We executed a two-step transfer learning method at varying proportions by combining two publicly available datasets, PVD and CPD. In this process, the initialization parameters of the feature extractor were transferred from Imagenet. Subsequently, we trained the feature extractor on the combined dataset and applied transfer learning to the final model with a DNN classifier for finetuning. The outcome is constructing a training model requiring fewer parameters for deployment on edge devices. Experiment results have demonstrated that our approach significantly improves the accuracy of plant leaf disease classification while maintaining efficiency in model parameter utilization. Our methodology holds potential for practical applications, such as aiding farmers in disease detection and control. Furthermore, we anticipate that our method will stimulate other researchers to explore novel machine-learning techniques to address real-world agricultural challenges.

REFERENCES

[1] Jo´zsef Popp, Ka´roly Peto˝, and Ja´nos Nagy. Pesticide productivity and food security. a review. Agronomy for sustainable development, 33:243– 255, 2013.

[2] Smriti Sharma, Rubaljot Kooner, and Ramesh Arora. Insect pests and crop losses. Breeding insect resistant crops for sustainable agriculture, pages 45–66, 2017.

[3] Jun Liu and Xuewei Wang. Plant diseases and pests detection based on deep learning: a review. Plant Methods, 17:1–18, 2021.

[4] Jun Liu and Xuewei Wang. Plant diseases and pests detection based on deep learning: a review. Plant Methods, 17:1–18, 2021.

[5] Neena Aloysius and M Geetha. A review on deep convolutional neural networks. In 2017 International Conference on Communication and Signal Processing (ICCSP), pages 0588–0592. IEEE, 2017.

[6] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. Proceedings of the IEEE, 107(8):1655–1674, 2019.

[7] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. Text data augmentation for deep learning. Journal of big Data, 8:1–34, 2021.

[8] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3d object detection for autonomous driving: A comprehensive survey. International Journal of Computer Vision, pages 1–55, 2023.

[9] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez- Paz. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2017.

[10] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. Journal of big data, 6(1):1–48, 2019.

[11] Faye Mohameth, Chen Bingcai, and Kane Amath Sada. Plant disease detection with deep learning and feature extraction using plant village. Journal of Computer and Communications, 8(6):10–22, 2020.

[12] Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. Plantdoc: A dataset for visual plant disease detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, pages 249–253. 2020.

[13] Guangzhou An, Hideo Yokota, Naohiro Motozawa, Seiji Takagi, Michiko Mandai, Shohei Kitahata, Yasukiko Hirami, Masayo Takahashi, Yasuo Kurimoto, and Masahiro Akiba. Deep learning classification models built with two-step transfer learning for age-related macular degeneration diagnosis. In 2019, the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pages 2049–2052, IEEE, 2019.

[14] Ruochi Zhang, Zhehao Guo, Yue Sun, Qi Lu, Zijian Xu, Zhaomin Yao, Meiyu Duan, Shuai Liu, Yanjiao Ren, Lan Huang, et al. Covid19xraynet: a two-step transfer learning model for the COVID-19 detecting problem based on a limited number of chest x-ray images. Interdisciplinary Sciences: Computational Life Sciences, 12:555–565, 2020.

[15] Raj Kumar, Anuradha Chug, Amit Prakash Singh, and Dinesh Singh. A systematic analysis of machine learning and deep learning based approaches for plant leaf disease classification: a review. Journal of Sensors, 2022, 2022.

[16] M Villaret et al. Grouped pointwise convolutions significantly reduce parameters in an efficient net. In Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence, IOS Press, volume 339, page 383, 2021.

[17] Joao Paulo Schwarz Schuler, Santiago Romani, Mohamed Abdel-Nasser, Hatem Rashwan, and Domenec Puig. Grouped pointwise convolutions reduce parameters in convolutional neural networks. In Mendel, vol- ume 28, pages 23–31, 2022.

[18] Joao Paulo Schwarz Schuler, Santiago Romani, Mohamed Abdel-Nasser, Hatem Rashwan, and Domenec Puig. Color-aware two-branch DCNN for efficient plant disease classification. In MENDEL, volume 28, pages 55–62, 2022.

[19] Hoang Trong Minh, Tuan Pham Anh, et al. A novel lightweight DCNN model for classifying plant diseases on Internet of things edge devices. In MENDEL, volume 28, pages 41–48, 2022.

MỘT MÔ HÌNH PHÁT HIỆN SÂU BỆNH HIỆU QUẢ TẠI BIÊN MẠNG SỬ DỤNG TẬP DỮ LIỆU LÀM GIÀU VÀ MẠNG NƠ RON TÍCH CHẬP

Tóm tắt- Cây trồng và sản lượng bị tổn hại đáng kể do bệnh cây trồng, đây là một trong những vấn đề nghiêm trọng nhất của nông nghiệp. Các nhà nghiên cứu gần đây đã nghiên cứu sử dụng trí tuệ nhân tạo (AI) để phát hiện và quản lý bệnh sớm một cách hiệu quả nhằm giải quyết vấn đề này. Nghiên cứu này tập trung phát triển phương pháp tối ưu hóa mô hình phân loại DCNN (Deep Convolutional Neural Network) đối với bệnh cây trồng. Chúng tôi đã làm phong phú dữ liệu bằng cách kết hợp dữ liệu từ hai bộ dữ liệu công khai, Bộ dữ liệu PlantVillage (PVD) và Bộ dữ liệu CroppedPlant (CPD) và chúng tôi đã đào tạo mô hình bằng cách sử dụng phương pháp học chuyển giao hai bước. Kết quả thực nghiệm chứng minh độ chính xác của mô hình là 82%, cao hơn nhiều so với các nghiên cứu trước đây. Đáng chú ý, việc đạt được kết quả này với ít tham số hơn trong khi vẫn duy trì hiệu suất phù hợp so với nghiên cứu trước đây chứng tỏ mô hình sử dụng hiệu quả các tài nguyên tính toán hạn chế. Do đó, mô hình đề xuất có thể triển khai trên các thiết bị biên để tối ưu hóa tính khả dụng và hiệu quả trong môi trường thế giới thực, đồng thời góp phần triển khai các dịch vụ nông nghiệp và điện toán biên mới.

Từ khoá- Bệnh lá, Tăng cường dữ liệu, Học chuyển giao, Điện toán biên**.**

Anh-Thu Pham received B.E degree of Telecommunication engineering from Posts and Telecommunications Institute of Technology (PTIT), Viet Nam, in 2003, and M.E degree of Telecommunication engineering from Royal Melbourne Institute of Technology, Australia, in 2008. She received the Ph.D. degree in Telecommunication engineering from PTIT, in 2010. Now, she is a lecturer in Telecommunication faculty of PTIT. Her research interests include networking, radio over fiber, and broadband networks..

Email: thupa@ptit.edu.vn

Thanh -Tra Nguyen received the Diploma of Engineer in Electronic Engineering and Telecommunication Technologies from Hanoi University of Science and Technology (HUST), and the Master of Engineering in Electronics and Telecommunications from Posts and Telecommunications Institute of Technology (PTIT) in 2000 and 2006, respectively. She is the senior lecturer at the Telecommunication Faculty 1 of PTIT.

Email: trant@ptit.ed.vn

Tuan-Anh Pham is currently a master's student in the Posts and Telecommunications Institute of Technology (PTIT). He received his bachelor's degree in Electronics and Telecommunications in 2022.
Email: anhpt.b22chte003@stu.ptit.edu.vn.

Van- Nhan Nguyen is currently a master's student in the Posts and Telecommunications Institute of Technology (PTIT). He received his bachelor's degree in Electronics and Telecommunications in 2022.
Email: nhannv.b18vt314@stu.ptit.edu.vn