Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 12, No. 2, 2024, 263-272.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

BNPL-Dataset: A New Benchmark Dataset for Visual Disease Detection of Barberry, Jujube, and Pomegranate Trees

Jalaluddin Zarei and Mohammad Hossein Khosravi*

Electrical and Computer Engineering Faculty, University of Birjand, Birjand, Iran.

Article Info

Article History:
Received 08 May 2024
Revised 10 August 2024
Accepted 25 August 2024

DOI: 10.22044/jadm.2024.14480.2552

Keywords:
Plant Disease Visual Dataset, Disease Detection, Barberry, Jujube, Pomegranate.

**Corresponding author*
mohokhosravi@birjand.ac.ir (M. H. Khosravi).

Abstract

Leaf diseases in agriculture can be challenging to detect in a timely manner due to factors such as lack of manpower, poor eyesight, and quarantine restrictions. To address this issue, convolutional neural networks (CNNs) are a promising solution. However, the performance of CNNs depends on large datasets, which are often scarce for local species.To address this problem, we introduce a new dataset, the "Birjand Native Plants Leaves (BNPL) Dataset," which contains images of healthy leaves and pests and diseases affecting three common plants in South Khorasan province: Barberry, Jujube, and Pomegranate. The dataset includes 9 classes, with a large volume of data, making it suitable for training CNNs.

We conducted experiments with several popular CNN architectures and gradient descent optimizers on the BNPL dataset. The results showed that the architectures, along with the optimizers, exhibited acceptable performance in classifying leaf diseases. Also, the BNPL dataset is publicly available to researchers[1].

1. Introduction

Agriculture is one of the most important means of livelihood in rural areas of Iran. The development of farms and orchards is the most important factor in the growth of the local economy, increasing the longevity in the villages, reducing migration and ultimately contributing to sustainable rural development. In addition to climate changes and drought, the increase in plant pests is also one of the most important concerns of farmers and gardeners. Early detection of tree diseases and action to eradicate pests helps to prevent the spread of the disease among other trees and minimizes possible damage.

In the past few decades, planting and maintaining jujube, barberry and pomegranate trees have become the most important agricultural and horticultural priorities in South Khorasan province, the easternmost province of Iran. Although the resistance of these trees to climate change and their little need for water have made them the first choice, their susceptibility to pests has become the most important challenge for gardeners. In doubtful cases, they usually use methods such as observing an expert or transferring the complication to the laboratory and conducting tests in diagnosis and following it. In addition to being time-consuming and expensive, these methods face challenges such as human factors, human error in diagnosis, or the limits of the transport.

Nowadays, machine vision based methods can identify the pests affecting the leaves in the shortest possible time and provide the possibility of eradicating the pests in time. But these methods need adequate images of leaves of objective trees in two healthy or diseased classes. In this research, we introduced the BNPL (Birjand Native Plants Leaves) dataset, a new benchmark dataset containing images of healthy and diseased leaves of barberry, jujube, and pomegranate trees. We

1. The BNPL dataset is available at the address https://kaggle.com/datasets/ec17162ca01825fb362419503cbc84c73d162bffe936952253ed522705228e06.

extend our dataset by augmenting methods with the aim of better training of models.

Our contributions are summarized as follows:

- To our knowledge, we build the first dataset containing different classes of Barberry, Jujube and Pomegranate pests. The whole dataset is made available to the research community.
- We conduct extensive experiments on the dataset using different CNN architectures, including VGG, ResNet, Inception, Xception, MobileNet and DenseNet, and establish the performance as the baseline for future research.

The rest of this paper is organized as follows. In Section 2, we review the related works concerning existing methods. Section 3 presents the methodology that introduces our dataset and the manner in which we collected its samples. The major aim of this research was to find the most suitable convolutional neural network that performs our classification task with the maximum performance. In Section 4, we present the experimental setup. In Section 5, we will examine the test conditions and the results of convolutional neural networks training on the proposed dataset.

2. Related works

In recent years, machine vision based techniques, especially methods based on deep networks, have shown acceptable performance in diagnosing plant complications [1]. Sanida et al. utilize twelve deep learning models in the agricultural field, and show the best result belongs to DensNet201 with 99.87% accuracy [2]. Similarly, in a study, Zhao and his colleagues gained 98.44, 99.43, and 95.20 percent accuracy by training the RIC-Net architecture on corn, potato, and tomato leaves, respectively [3].

In another study, Singh et al. considered 1363 infested images and 929 healthy images of corn as input to the AlexNet model, finally reaching 99.16% accuracy [4]. Taking another look at the comparison of three CNN such as GoogleNet, VGGNet-16, and EfficientNet-B0, used for the detection of potato leaf rust by Mackenzie Gapsil et al., it can be seen that the EfficientNet architecture provides better performance than the other two architectures [5]. In a study by Attila et al., versions B0 to B7 of the EfficientNet architecture were trained on two original and augmented datasets with 55448 and 61486 images, respectively. Finally, their accuracy compares with AlexNet, ResNet50, VGG16, and Inception V3 architectures. Results indicate that B5 and B4 versions have the highest accuracy compared to other architectures [6]. Al-Hassouni et al. utilize MobileNet architecture on smartphones to detect tomato leaf diseases. This architecture is employed to overcome the limited computing resources and energy of smartphones [7].

In another research, Najafabadi et al. introduce AgriNet architecture with different motivations [8]. In AgriNet, researchers have used the squeeze operation of SENet and the shuffling operation of ShuffleNet architectures.

2.1. Related Datasets

The major concern in employing architecturs based on deep neural networks is necessity of massive datasets and the dependence of the models on these input datasets [9]. Many researches have been done on the production of rich image datasets in this field [10]. Different and diverse datasets have been introduced in the field of disease and plant pest diagnosis in recent years. Table 1 shows some of the most popular datasets in this field.

Table 1. Plant species in different datasets [11-17].

No	Dataset	Plant species	Number of images	Number of classes
1	PlantVillage	Apple, Blueberriy, Cherry, Corn, Grape, Grange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato	54309	38
2	PlantDoc	Apple, Bell Pepper, Blueberriy, Cherry, Corn, Grape, Peach, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato	2598	17
3	Rice Leaf Diseases	Rice	120	3
4	Plant Disease Symptoms (PDDB)	Black pepper, Common Bean, Cashew tree, Cassava, Citrus, Cocos nucifera, Coffee, Corn, Cotton, Cupuaçu, Grapevines, Kale, Melon, Oil Palm, Papaya, Passion fruit, Pinneaple, Rice, Soybean, Sugarcane, Wheat	2326	171
5	Northern Leaf Blight (NLB)	Corn	18222	1
6	IP102: Insect Pest Recognition	Rice, Corn, Wheat, Beet, Alfalfa, Vitis, Citru, Mango	75000	102
7	Fieldplant	Cassava, Corn, Tomato	8629	27

Despite the various plant image datasets that have been created in this research field, as far as we know, no dataset containing images of leaves of native trees of South Khorasan province has been introduced. Here, a new dataset containing images of healthy and diseased leaves of barberry, jujube, and pomegranate trees has been introduced. These plants have a strategic position in the agricultural sector of Iran, especially in South Khorasan province. So, creating this dataset is of great importance in diagnosing and timely combating pests and diseases of these plants.

3. Proposed dataset: Birjand Native Plants Leaves (BNPL) Dataset

3.1. Methodology

To create BNPL dataset, we collected leaves of barberry, jujube, and pomegranate trees in the south of Khorasan. These trees are native to South Khorasan province, and their products are the main contribution to the agricultural production and economy of the province. Hence, preparing a dataset from healthy and diseased leaves can be very useful for fast diagnosis and controlling pests and diseases.

A detailed workflow of our dataset creation and examination is depicted in Fig 1. Based on this workflow, our work was done in two stages: first, gathering, cleaning, labeling and post-processing relevant images to build the target dataset, and second, employing the various pre-trained CNN networks and retrain their dense classification layers to examine their performance on the proposed dataset. This approach will propose a suitable architecture with both accuracy and reliability for easy adoption and implementation in various real agricultural scenarios.

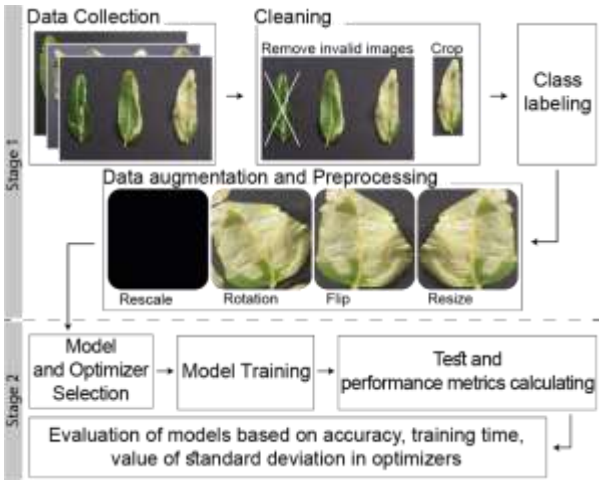


Figure 1. Workflow of our Proposed Method

3.2. Categories of BNPL

We collected 4293 different samples of leaves of barberry, jujube, and pomegranate trees in the following categories, which are illustrated in Figure 2:

- Healthy Barberry leaves: are healthy in appearance and have a bright green color.
- Barberry rust disease: Yellow or orange blisters appear on the leaves due to the growth of a type of fungus. Wind plays an effective role in displacing this disease, and by increasing the damage, it causes the shrub to decline [18].
- Barberry Pandemis ribeana Tortricidae pest: The larva of this insect feeds on the leaf tissue and the leaves take on a net shape, which causes a decrease in the greenness of the leaf surface, followed by a decrease in food production and a weakening of the shrub [18].
- Healthy Jujube leaves: The leaves of this category are in different sizes with a healthy appearance and bright green color.
- Jujube Ziziphus Tingid disease: This pest feeds on the cell sap on the back of the leaves, which causes pale areas to appear on the leaves, and the effects of the pest's excreta appear as black dots on the back of the leaves. By feeding on leaf sap, the pest weakens the growth of leaves and eventually reduces the growth of trees [19].
- Jujube Parenchyma-Eating Butterfly pest: This butterfly feeds on the leaf tissue and reduces the production of leaf photosynthetic materials, tree growth, and fruit weight gain.
- Healthy Pomegranate leaves: Like the other healthy leaves mentioned in this dataset, this group of leaves is also correct in appearance and has their natural color (light green and close to dark green).
- Pomegranate Aphis punicae pest: This pest feeds on leaf sap, then with honeydew secretion, it reduces leaf growth, weakens the tree and ultimately reduces tree production.
- Pomegranate Leaf-Cutting Bees pest (Mega-Chile): By cutting pieces of the leaf, they separate regular pieces from the edge of the leaf and finally use these pieces to build a nest. This bee isn't known as a pest due to its low damage. But as soon as the symptoms are observed, it is better to take preventive measures.

3.3. Image Collection

All Samples of the proposed dataset were taken using a smartphone camera (without any flash-light) from the front and back of the leaves of the target trees between 9:00 AM and 6:00 PM in natural sunlight in different gardens. Also, no digital or optical zooming was used during shooting. Finally, the images were cropped to include only the leaves and saved in the appropriate size.

3.4. Data augmentation and Preprocessing methods

The dataset used in this research consists of 4293 images classified into nine classes. Data augmentation techniques can increase the volume of the dataset and then overcome overfitting and improve generalization [2, 20]. These techniques create new samples by performing artificial transformations on the dataset and cause better performance of convolutional neural networks [21].

There are many techniques to increase the data, such as resizing, rotation, cropping, contrast adjustment, flipping, etc [21]. We used rotation by $0.1 \times 2\pi$ degrees and a horizontal flip on the train part.

In the preprocessing stage, all images are scaled by 1\.255 and resized to fit the desired convolutional neural networks (256×256 pixels).


a) Healthy Barberry leaves (number of samples: 509)


b) Barberry Rust disease (number of samples: 384)


c) Barberry Pandemis ribeana Tortricidae pest (number of samples: 536)


d) Healthy Jujube leaves (number of samples: 522)


e) Jujube Ziziphus Tingid disease (number of samples: 517)


f) Jujube Parenchyma-Eating Butterfly pest (number of samples: 623)


g) Healthy Pomegranate leaves (number of samples: 418)


h) Pomegranate Aphis punicae pest (number of samples: 184)


i) Pomegranate Leaf-Cutting Bees pest (number of samples: 576)

Figure 2. Sample images of the types of images available in the proposed dataset.

4. Benchmarking BNPL Dataset

In this section, we evaluate the performance of state-of-the-art deep convolutional networks on the BNPL dataset, including VGGNet-19, ResNet, Inception, Xception, MobileNet and DenseNet.

4.1. Experimental Setup

We use the resources and tools of the Kaggle data science community platform with the hardware requirements mentioned in Table 2. Also, we used Python with Keras 2.8.0.

Table 2. Hardware requirements of the Kaggle server.

Requirement	Specification
Processor	Intel(R) Xeon® CPU @ 2.00 GHz
RAM	13 GB
Hard Disk	73.1 GB - SSD
Video Card	Nvidia Tesla P100 GPU
GPU	16 GB

4.2. Training

Our dataset was divided randomly into training, validation, and test sets with 80%, 10%, and 10%, respectively. The training and validation sets are only used for training and fitting the model, while the test set is used to evaluate the prediction performance on samples that weren't used in the training process.

The architectures mentioned in this article use the transfer learning approach. In this approach, CNN models are pre-trained on the ImageNet dataset, and then the obtained weights effectively speed up the learning phase and reduce the need for many samples [22]. In deep learning, upper layers (convolutional layers) are used to identify features. Meanwhile, in transfer learning, the convolutional layers are kept unchanged, and the last few layers of the network (output layers) are replaced with new layers and retrained on the new dataset.

Softmax was chosen as the activation function in the last layer, and the loss function was selected as Categorical Cross entropy.

The model-optimization technique is significant in determining the performance of the CNN model. Here, we use the gradient descent optimizers for each model mentioned above. In addition, we set the learning rate as $1 \times e^{-5}$ for all optimization methods. The models train in 15 epochs.

4.3. Performance metrics

The performance evaluation of the presented

models is calculated based on performance metrics of Accuracy, Precision, Recall, and F1-Score, as below, in which, TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{Precision + Recall} \quad (4)$$

Accuracy measures the accuracy of correctly recognized samples. Precision estimates the correctness rate of the TP relative to its performance (prediction). Unlike the Precision metric, Recall calculates the rate of TP in actuality. F1-Score calculates the geometric mean of Precision and Recall.

Each model is trained with all the gradient descent optimizers. Different metrics such as accuracy, precision, sensitivity, and F1-Score will be calculated separately for each model and optimizer. Finally, the best model with an optimizer algorithm is selected. In multi-class problems, the value of each metric is calculated for each of the classes separately, and then their average forms the value of the desired metrics.

5. Results and discussion

The main goal of this research is to introduce the BNPL-Dataset and examine the performance of state-of-the-art deep learning architectures on it. By calculating and comparing the accuracy, precision, recall, and f1-Score we evaluate the performance of these models. Also, all models use the gradient descent optimizer algorithms, namely, SGD, AdaGrad, AdaDelta, RMSprop, AdaM, AdaMax, and Nadam.

Tables 3 to 5 show the performance results of the architectures based on the above performance metrics, training time and the standard deviation of the architecture in the optimizers.

In Table 3, in each architecture, the highest value obtained in one of the four evaluation metrics is bolded. The purpose of doing this is to determine the best optimizer function in each model.

Table 3. Performance of trained models on the BNPL.

		Ac (%)	Pr (%)	Re (%)	F1 (%)
VGG19	SGD	97.99	97.98	97.54	97.76
	Adagrad	93.97	95.61	92.41	93.98
	Adadelta	36.38	0	0	0
	RMSprop	98.66	98.66	98.66	98.66
	Adam	99.33	99.33	99.33	99.33
	Adamax	99.33	99.33	99.33	99.33
	Nadam	**99.78**	**99.78**	**99.78**	**99.78**
Standard deviation		1.97	1.39	2.53	1.97
ResNet152V2	SGD	98.88	98.88	98.88	98.88
	Adagrad	99.33	99.33	99.11	99.22
	Adadelta	57.81	91.75	19.87	32.66
	RMSprop	96.65	96.65	96.65	96.65
	Adam	**100**	**100**	**100**	**100**
	Adamax	**100**	**100**	**100**	**100**
	Nadam	99.78	99.78	99.78	99.78
Standard deviation		1.17	1.17	1.16	1.17
InceptionV3	SGD	99.55	99.55	99.55	99.55
	Adagrad	98.88	99.10	98.44	98.77
	Adadelta	71.43	98.78	18.08	30.57
	RMSprop	97.10	97.53	97.10	97.31
	Adam	99.11	99.33	99.11	99.22
	Adamax	**99.78**	**99.78**	**99.78**	**99.78**
	Nadam	99.55	99.55	99.55	99.55
Standard deviation		0.90	0.75	0.92	0.83
Inception ResNetV2	SGD	97.77	98.20	97.54	97.87
	Adagrad	97.77	98.42	97.10	97.75
	Adadelta	74.33	96.10	16.52	28.19
	RMSprop	**99.78**	**99.78**	**99.55**	**99.66**
	Adam	97.54	97.54	97.54	97.54
	Adamax	98.88	98.88	98.88	98.88
	Nadam	99.55	**99.78**	**99.55**	**99.66**
Standard deviation		0.90	0.82	1.00	0.88
Xception	SGD	98.44	98.44	98.44	98.44
	Adagrad	98.88	98.88	98.88	98.88
	Adadelta	65.40	93.20	21.43	34.85
	RMSprop	**99.55**	**99.55**	**99.33**	**99.44**
	Adam	98.88	98.88	98.88	98.88
	Adamax	98.66	98.88	98.66	98.77
	Nadam	98.88	98.88	98.88	98.88
Standard deviation		0.34	0.32	0.27	0.29
MobileNetV2	SGD	99.33	99.33	99.33	99.33
	Adagrad	99.11	99.11	98.88	98.99
	Adadelta	52.90	87.50	9.38	16.94
	RMSprop	**99.55**	**99.55**	**99.55**	**99.55**
	Adam	98.21	98.21	98.21	98.21
	Adamax	98.88	98.88	98.88	98.88
	Nadam	98.66	98.66	98.66	98.66
Standard deviation		0.44	0.44	0.44	0.44
DenseNet201	SGD	99.33	99.33	99.33	99.33
	Adagrad	99.33	99.33	99.33	99.33
	Adadelta	61.16	100	8.48	15.63
	RMSprop	99.11	99.11	99.11	99.11
	Adam	**99.55**	**99.55**	**99.55**	**99.55**
	Adamax	**99.55**	**99.55**	**99.55**	**99.55**
	Nadam	99.33	99.33	99.33	99.33
Standard deviation		0.15	0.15	0.15	0.15

Based on Table 3, we can get the following results:

- The best result for optimizer functions is related to Adam and Adamax algorithms with a value of 100% (in all performance metrics) in ResNet152V2 architecture. This good result can be related to the innovation of the ResNet architecture, which is "skip" connections. The

skip connection adds the input of a residual block directly to the output of the same block and improves the challenges caused by the vanishing gradient.

- Respectively, ResNet152V2, InceptionResNetV2, and DenseNet201 have obtained the best results in performance metrics in two optimization algorithms. This result shows that these architectures are more flexible than others in working with gradient reduction optimizers.
- Adam, Adamax, RMSprop, and Nadam optimizers have performed better on the dataset.
- All convolutional neural network architectures with gradient descent optimization algorithms on the proposed dataset have shown acceptable performance. However, the performance of the Adadelta optimizer in all architectures has had weak and unstable results. Therefore, we ignore the results of this optimizer in the subsequent reviews.

Table 4. Estimated training time of architectures in each gradient descent optimization algorithm(in minutes).

	SGD	Adagrad	RMSprop	Adam	Adamax	Nadam	Sum Total
MobileNetV2	1.33	1.33	1.33	1.33	1.33	1.40	**8.07**
InceptionV3	2.10	2.12	2.08	2.08	2.12	2.13	**12.63**
VGG19	3.08	3.05	3.05	3.05	3.05	3.05	18.33
Xception	3.10	3.08	3.08	3.10	3.08	3.10	18.55
DenseNet201	3.80	3.53	3.55	3.52	3.53	3.78	21.72
InceptionResNetV2	4.28	4.30	4.27	4.27	4.25	4.27	25.63
ResNet152V2	5.50	5.50	5.45	5.50	5.45	5.47	32.87

Table 4. shows the time spent training the mentioned architectures in each optimization algorithm. If we consider the time performance of each architecture equal to the total time spent in its optimizers, So we can get the following results:

- MobileNetV2 and ResNet152V2 spend the least and the most time training in the proposed dataset, respectively. By comparing the structure of these architectures, it can be considered the stacked residual units in the Resnet152v2 architecture cause more time.
- According to Table 3, ResNet152V2 has the best performance in accuracy metric with a value of 100%, followed by InceptionResNetV2 with a value of 99.78% and DenseNet201 with a value of 99.55%. On the other hand, by comparing the time performance of these three architectures, we realize that they need to spend more time on training.
- Since the MobileNetV2 and InceptionV3 architectures are in the category of Lightweight architectures, they need fewer resources and time to perform their calculations. This point is consistent with the results obtained in Table 4. On the other hand, MobileNetV2 and InceptionV3 show acceptable results in terms of efficiency, in contrast to spending less time and resources.

Table 5. Standard deviation of the accuracy metric of gradient descent optimizers for each architecture.

	Standard deviation of Accuracy in all optimizers
DenseNet201	**0.15**
Xception	0.34
MobileNetV2	0.44
InceptionV3	0.90
InceptionResNetV2	0.90
ResNet152V2	1.17
VGG19	1.97

In Table 5. we have considered the accuracy values in SGD, RMSprop, Adam, Adagrad, Adamax, and Nadam optimizers for each architecture. Then, we calculate their standard deviation and record the obtained value for the relevant architecture.

The goal is to determine which architectural structure has a more stable behavior in working with gradient descent optimizers. In other words, which architecture has more independence from optimizer algorithms to achieve a desired result. The dense structure of DenseNet201 has shown more stable behavior in this study. In contrast, VGG19 has low stability (You can also see standard deviation values of other performance metrics of this architecture in Table 3).

By examining all the conclusions made, we can see that ResNet152V2, InceptionResNetV2, and DenseNet201 algorithms have the best results based on the performance evaluation metrics in different optimizers, respectively. Considering their time performance, DenseNet201 has the best time performance in addition to the best performance. DenseNet201 also has the best standard deviation in addition to the best efficiency and time performance.

Next, we move on to the confusion matrices. In confusion matrices, we first show the efficiency of the ResNet152V2 architecture in different optimizers and then the efficiency of the Adam optimizer in various architectures. Based on the results, ResNet152V2 and Adam were the best model and optimizer algorithm, respectively.

Figures 3 and 4 show the confusion matrices related to the ResNet152V2 model in all optimizers and the confusion matrices of the models after testing with the Adam optimizer, respectively. These confusion matrices show that the models presented in this study have a small error in choosing the correct samples. If the number of FP and FN samples outside the main diagonal is less, it indicates the better performance of the model. On the other hand, the numbers on the main diagonal indicate the number of samples that are correctly classified in their respective class.

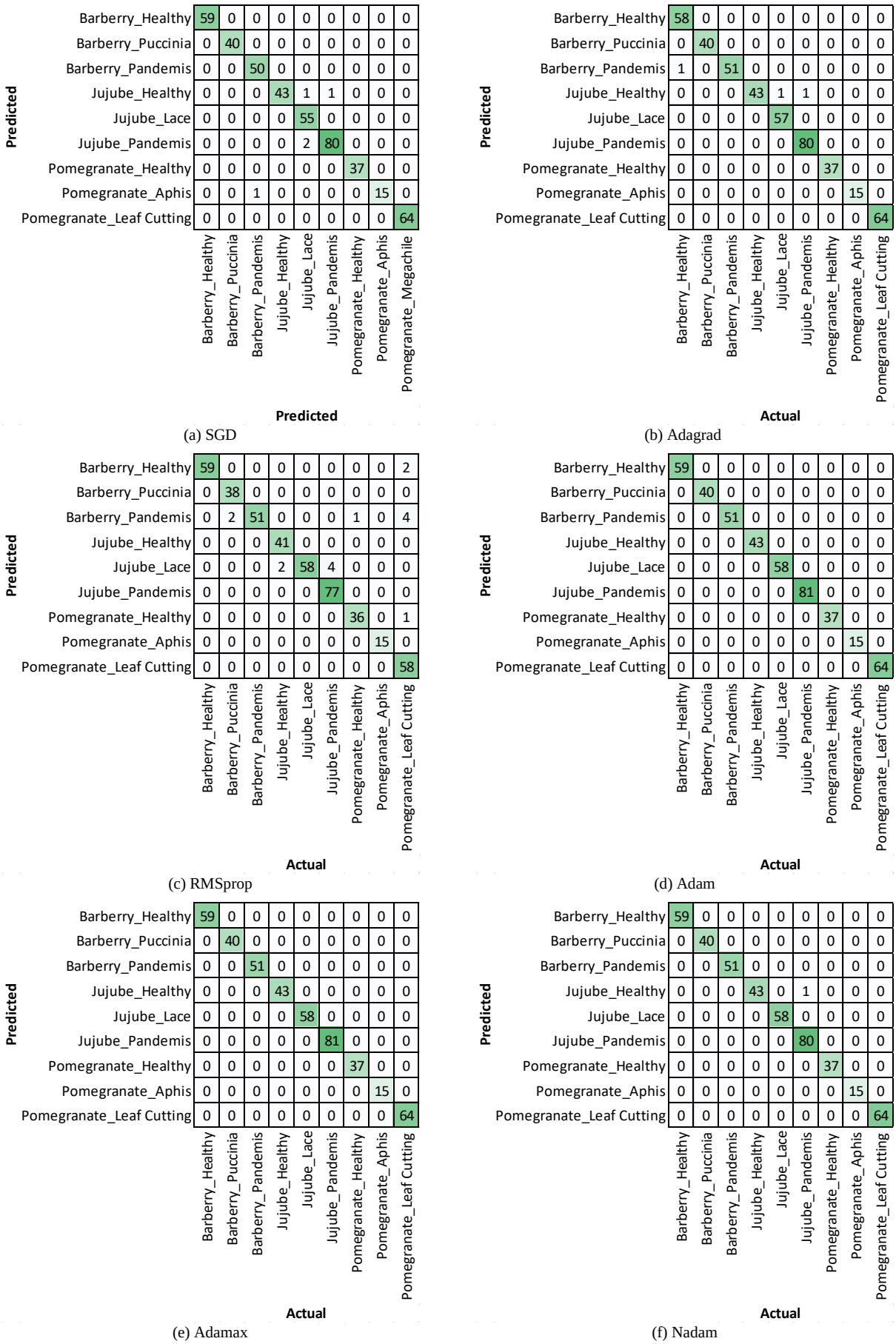
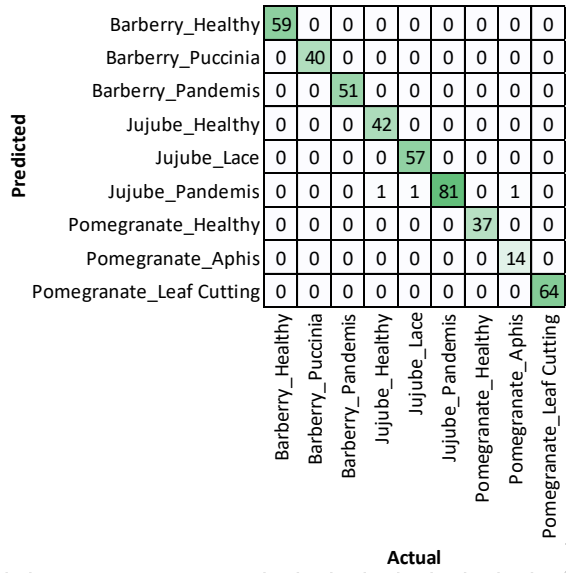
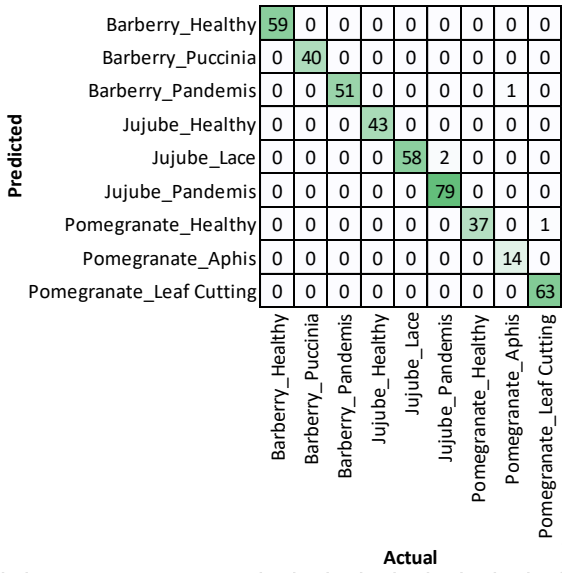


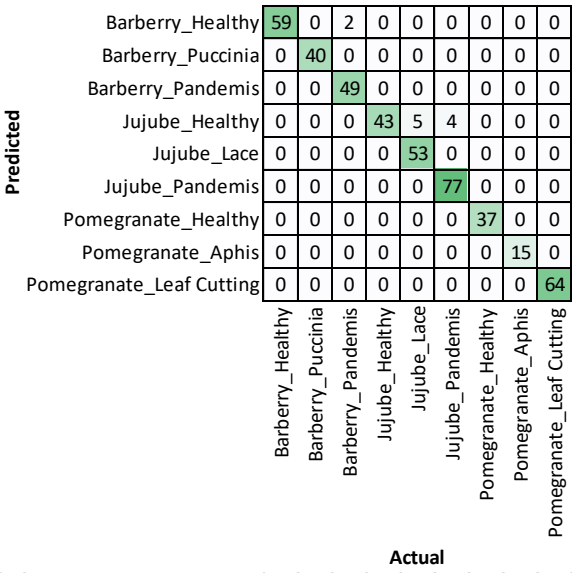
Figure 3. Confusion matrices related to ResNet152V2 in gradient descent optimizers



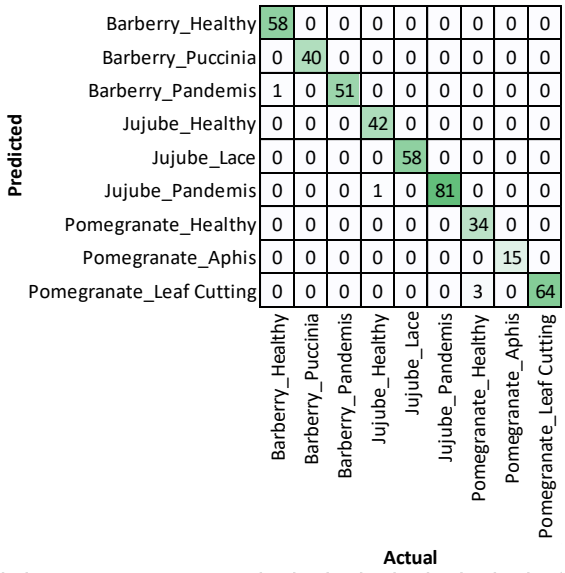
(a) VGG19



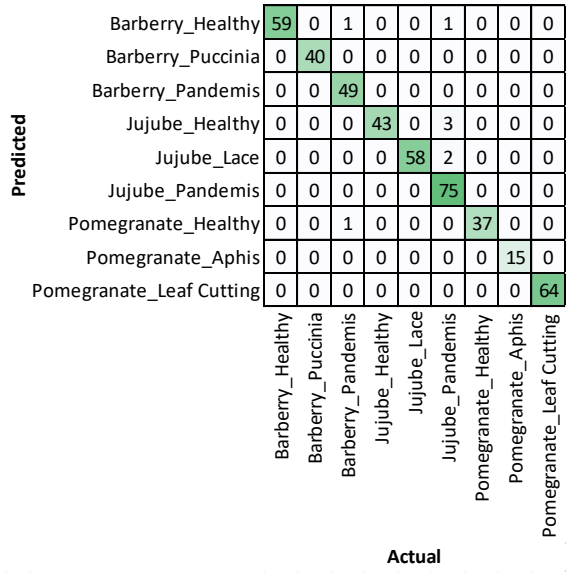
(b) InceptionV3



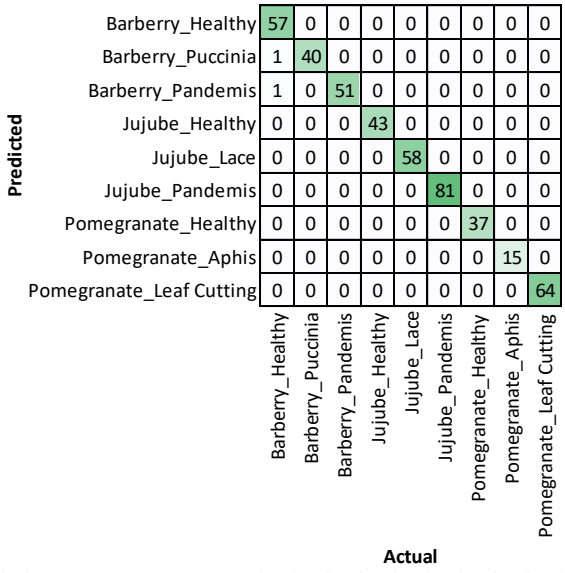
(c) InceptionResNetV2



(d) Xception



(e) MobileNetV2



(f) DenseNet201

Figure 4. Confusion matrices related to the models with Adam optimizer

6. Conclusions

In this work, we collect a plant dataset, named BNPL, for disease and pest recognition of native plants of Southern Khorasan province, namely, barberry, jujube, and pomegranate. This dataset consists of 4293 images classified into nine classes (Healthy Barberry leaves, Barberry Rust disease, Barberry Pandemis ribeana Tortricidae pest, Healthy Jujube leaves, Jujube Ziziphus Tingid disease, Jujube Parenchyma-Eating Butterfly pest, Healthy Pomegranate leaves, Pomegranate Aphis punicae pest, and Pomegranate Leaf-Cutting Bees pest). In addition, we conducted a massive classification task on this new dataset using seven state-of-the-art CNN architectures and found that ResNet152V2 has achieved the best performance by Adam and Adamax optimizers with 100% accuracy, DenseNet201 with 99.55% accuracy and 0.15 standard deviation for its performance in gradient descent optimizers has the minimum dependence on the structure of gradient descent optimizers, MobileNetV2 and InceptionV3 have the best performance with 8.07 and 12.63 minutes and 99.78 and 99.55 percent accuracy, respectively, and also, DenseNet201 has the best performance considering the criteria of accuracy, standard deviation and time together.

In the upcoming works, we want to collect other image datasets from the leaves of other native trees of the province that have a great contribution to the economy, food security, and environmental protection. On the other hand, creating and sharing datasets with appropriate recommendations in an international system can help identify and prevent pests and diseases quickly throughout the world.

References

[1] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," *Computers and Electronics in Agriculture*, vol. 154, pp. 18-24, 2018.

[2] T. Sanida, D. Tsiktsiris, A. Sideris, and M. Dasygenis, "A heterogeneous implementation for plant disease identification using deep learning," *Multimedia Tools and Applications*, vol. 81, no. 11, pp. 15041-15059, 2022.

[3] Y. Zhao, C. Sun, X. Xu, and J. Chen, "RIC-Net: A plant disease classification model based on the fusion of Inception and residual structure and embedded attention mechanism," *Computers and Electronics in Agriculture*, vol. 193, no. 2, p. 106644, 2022.

[4] R. K. Singh, A. Tiwari, and R. K. Gupta, "Deep transfer modeling for classification of Maize Plant Leaf Disease," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6051-6067, 2022.

[5] H. Afzaal, A. A. Farooque, A. W. Schumann, N. Hussain, A. McKenzie-Gopsill, T. Esau, F. Abbas and B. Acharya, "Detection of a Potato Disease (Early Blight) Using Artificial Intelligence," *Remote Sensing*, vol. 13, no. 3, p. 411, 2021.

[6] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecological Informatics*, vol. 61, no. 1, p. 101182, 2021.

[7] A. Elhassouny and F. Smarandache, "Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks," in *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, pp. 1-4, 2019.

[8] F. Salimian Najafabadi and M. T. Sadeghi, "AgriNet: a New Classifying Convolutional Neural Network for Detecting Agricultural Products' Diseases," *Journal of AI and Data Mining*, vol. 10, no. 2, pp. 285-302, 2022.

[9] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Computers and Electronics in Agriculture*, vol. 187, no. 8, p. 106279, 2021.

[10] A. Dey and S. Biswas, "Shot-ViT: Cricket Batting Shots Classification with Vision Transformer Network," *International Journal of Engineering*, vol. 37, no. 12, pp. 2463-2472, 2024.

[11] D. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.

[12] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: a dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pp. 249-253, 2020.

[13] H. B. Prajapati, J. P. Shah, and V. K. Dabhi, "Detection and classification of rice plant diseases," *Intelligent Decision Technologies*, vol. 11, no. 3, pp. 357-373, 2017.

[14] T. Wiesner-Hanks, E. L. Stewart, N. Kaczmar, C. DeChant, H. Wu, R. J. Nelson, H. Lipson and M. A. Gore, "Image set for deep learning: field images of maize annotated with disease symptoms," *BMC research notes*, vol. 11, no. 1, pp. 1-3, 2018.

[15] J. G. A. Barbedo, L. V. Koenigkan, B. A. Halfeld-Vieira, R. V. Costa, K. L. Nechet, C. V. Godoy, M. L. Junior, F. R. A. Patricio, V. Talamini, L. G. Chitarra, S. A. S. Oliveira, A. K. N. Ishida, J. M. C. Fernandes, T. T. Santos, F. R. Cavalcanti, D. Terao and F. Angelotti, "Annotated Plant Pathology Databases for Image-Based Detection and Recognition of Diseases," *IEEE Latin America Transactions*, vol. 16, no. 6, pp. 1749-1757, 2018.

[16] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *Proceedings of the*

IEEE/CVF conference on computer vision and pattern recognition, pp. 8787-8796, 2019.

[17] E. Moupojou, A. Tagne, F. Retraint, A. Tadonkemwa, D. Wilfried, H. Tapamo and M. Nkenlifack, "FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning," *IEEE Access*, vol. 11, pp. 35398-35410, 2023.

[18] H. Zeraatgar, G. Tavakkoli-Korghond, S. Z. Mousavi, N. Pourfateh, "Seedless barberry, limitations and damaging factors," *Agricultural Research, Education and Extension Organization (AREEO)*, 2017. (In Persian)

[19] G. Tavakkoli-Korghond, H. Zeraatgar, K. Ghos, M. Yousefi, A. R. Rezaei-Gazik, M. R. Mirzaee, " Jujube Ziziphus Tingid, preliminary knowledge and available experiences for its control, " *Journal of barberry and jujube*, vol. 3, no. 1, pp. 22-29, 2021. (In Persian)

[20] V. Gonzalez-Huitron, J. A. León-Borges, A. E. Rodriguez-Mata, L. E. Amabilis-Sosa, B. Ramírez-Pereda, and H. Rodriguez, "Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4," *Computers and Electronics in Agriculture*, vol. 181, no. 2, p. 105951, 2021.

[21] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," *Recent trends and advances in artificial intelligence and Internet of Things*, pp. 519-567, 2020.

[22] S. M. R. Hashemi, H. Hassanpour, E. Kozegar, and T. Tan, "Cystoscopic image classification by unsupervised feature learning and fusion of classifiers," *IEEE Access*, vol. 9, pp. 126610-126622, 2021.

BNPL-Dataset: یک مجموعه داده معیار جدید برای تشخیص بصری بیماری‌های درختان زرشک، عناب و انار

جلال‌الدین زارعی و محمدحسین خسروی*

دانشکده مهندسی برق و کامپیوتر، دانشگاه بیرجند، بیرجند، ایران.

ارسال ۲۰۲۴/۰۵/۰۸؛ بازنگری ۲۰۲۴/۰۸/۱۰؛ پذیرش ۲۰۲۴/۰۸/۲۵

چکیده:

تشخیص به‌موقع بیماری‌های برگ درختان در کشاورزی به دلیل عواملی مانند کمبود نیروی انسانی، ضعف بینایی و محدودیت‌های قرنطینه‌ای می‌تواند با چالش‌هایی همراه باشد. برای حل این چالش‌ها استفاده از روش‌های مبتنی بر یادگیری ماشین، و بطور خاص بکارگیری شبکه‌های عصبی کانولوشنی (CNNs) امیدوارکننده است. با این حال، عملکرد CNNها به مجموعه داده‌های بزرگی بستگی دارد که اغلب برای گونه‌های محلی کمیاب هستند. برای رفع این مشکل، مجموعه داده جدیدی به نام «مجموعه داده برگ گیاهان بومی بیرجند (BNPL-Dataset)» را معرفی می‌کنیم که حاوی تصاویری از برگ‌های سالم، بیمار و آفت‌زده از سه درخت بومی در استان خراسان جنوبی به نام‌های زرشک، عناب و انار است. این مجموعه داده شامل ۹ کلاس با حجم زیادی از داده است که آن را برای استفاده در آموزش شبکه‌های عصبی عمیق کانولوشنی مناسب می‌کند. در این پژوهش آزمایش‌هایی با چندین معماری رایج شبکه‌های عصبی عمیق کانولوشنی و بهینه‌سازهای گرادیان کاهشی روی مجموعه داده BNPL انجام شد. نتایج نشان می‌دهند که معماری‌های آزمون شده بر روی مجموعه داده پیشنهادی عملکرد قابل قبولی در طبقه‌بندی بیماری‌های برگ داشته‌اند. در این میان معماری ResNet150 بهترین عملکرد را از خود نشان می‌دهد. مجموعه داده BNPL به صورت عمومی در دسترس محققان قرار گرفته است.[1]

کلمات کلیدی: مجموعه داده بصری بیماری‌های گیاهی، تشخیص بیماری، زرشک، عناب، انار.

۱. BNPL-Dataset از طریق آدرس زیر قابل دسترس است:
https://kaggle.com/datasets/ec17162ca01825fb362419503cbc84c73d162bffe936952253ed522705228e06