

Article

Plant Image Classification with Nonlinear Motion Deblurring Based on Deep Learning

Ganbayar Batchuluun, Jin Seong Hong , Abdul Wahid  and Kang Ryoung Park *

Division of Electronics and Electrical Engineering, Dongguk University, 30 Pildong-Ro, 1-Gil, Jung-Gu, Seoul 04620, Republic of Korea; ganabata87@dongguk.edu (G.B.); turtle1990@dgu.ac.kr (J.S.H.); abdulwahid9383@dgu.ac.kr (A.W.)

* Correspondence: parkgr@dongguk.edu

Abstract: Despite the significant number of classification studies conducted using plant images, studies on nonlinear motion blur are limited. In general, motion blur results from movements of the hands of a person holding a camera for capturing plant images, or when the plant moves owing to wind while the camera is stationary. When these two cases occur simultaneously, nonlinear motion blur is highly probable. Therefore, a novel deep learning-based classification method applied on plant images with various nonlinear motion blurs is proposed. In addition, this study proposes a generative adversarial network-based method to reduce nonlinear motion blur; accordingly, the method is explored for improving classification performance. Herein, experiments are conducted using a self-collected visible light images dataset. Evidently, nonlinear motion deblurring results in a structural similarity index measure (SSIM) of 73.1 and a peak signal-to-noise ratio (PSNR) of 21.55, whereas plant classification results in a top-1 accuracy of 90.09% and F1-score of 84.84%. In addition, the experiment conducted using two types of open datasets resulted in PSNRs of 20.84 and 21.02 and SSIMs of 72.96 and 72.86, respectively. The proposed method of plant classification results in top-1 accuracies of 89.79% and 82.21% and F1-scores of 84% and 76.52%, respectively. Thus, the proposed network produces higher accuracies than the existing state-of-the-art methods.

Keywords: nonlinear motion; motion deblurring; deep learning; plant image classification; generative adversarial network

MSC: 68T07; 68U10



Citation: Batchuluun, G.; Hong, J.S.; Wahid, A.; Park, K.R. Plant Image Classification with Nonlinear Motion Deblurring Based on Deep Learning. *Mathematics* **2023**, *11*, 4011. <https://doi.org/10.3390/math11184011>

Academic Editors: Hongang Qi, Yan Liu and Jun Miao

Received: 30 August 2023

Revised: 14 September 2023

Accepted: 19 September 2023

Published: 21 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In this study, we conducted two different experiments such as an image deblurring and a classification. In the image deblurring method, nonlinear motion blur is considered. First, we reduce blurring from images which occurred in a nonlinear way. Second, the images restored via the image deblurring method are classified into 28 classes via the classification method. In addition, plant (flower and leaf) images were used in this study. The total number of classes of the plant images is 28.

Moreover, there have been previous studies on classification applied to plant images [1–9]. In image-based studies, classification can be performed by extracting the color and pattern information of plants from the acquired images. However, the extraction of such information when motion blur occurs in an image is slightly challenging. Motion blur can occur via plant movements owing to wind or via the movements of a camera as the hand holding the camera may shake. Furthermore, extracting color and pattern information of plants becomes extremely challenging when both these movements occur simultaneously in an image. Generally, natural motion blur is nonlinear. However, studies on plant image-based classifications considering nonlinear motion blur are limited. Thus, this study proposed a plant image-based nonlinear motion deblurring method and examined the improvement

in plant image classification through the proposed deblurring methods. A deep learning method named the “generative adversarial network” (GAN) [10] is used in our plant image-based nonlinear motion deblurring method (PI-NMD), whereas another deep learning method named “convolutional neural network” (CNN) [11] based on residual blocks was used in our plant image-based classification method (PI-Clas). The details of our methods are described in Section 3. Our contributions are explained in the following:

- Numerous studies on plant image-based research have been conducted; however, plant image classification studies considering nonlinear motion blur are limited. This is the first study to perform plant image classification considering nonlinear motion blur.
- This study newly proposed the novel PI-NMD network, in which a blurred image ($300 \times 300 \times 3$) obtained by generating nonlinear motion blur is used as an input. Our PI-NMD network uses nine residual blocks and generates a deblurred image based on the residual features in the blurred image.
- This study proposed the novel PI-Clas network in which a deblurred image is used as an input. Our PI-Clas network uses 12 residual blocks and performs classification using the image restored based on residual features.
- The proposed PI-NMD and PI-Clas models, as well as the visible light plant image database, are disclosed through [12] to be used by other researchers.

The rest of the paper is organized as follows. Section 2 presents detailed explanations of previous plant image-based studies including classification, deblurring, and segmentation-based methods. Section 3 explains the PI-NMD and PI-Clas methods in detail, in which layers of structures are explained with tables and figures. Section 4 presents the experimental results including ablation studies, testing, training, and comparisons. In Section 5, a discussion of the study is provided with error cases of the methods. Finally, the conclusion is provided in Section 6.

2. Related Works

2.1. Plant Image-Based Classification Methods

Herein, we review existing classification research based on plant images. A previous work on rice yield image classification [1] utilized an unmanned aerial vehicle (UAV) platform with a hyperspectral camera and the XGBoost algorithm. It performed a classification based on an image dataset acquired using UAV and utilized intensity saturation images as inputs in the training and testing phases. Another study on soybean yield image classification [2] utilized support vector machine (SVM) with a radial basis function (SVMR) and a drone equipped with a camera. Here, the images acquired using a drone and camera were classified into three classes, gray, light tawny, and tawny pubescence. Moreover, a study on plant and plant disease classification [3] conducted various experiments on the PlantDoc database [13] and proposed a novel attention-augmented residual (AAR) network. The proposed AAR included a stacked pre-activated residual block and attention block, which extract deep coarse-level features and salient feature sets, respectively. Additionally, another study conducted on the PlantDoc database [4] utilized a DenseNet-121 model, in addition to a lightweight architecture and Fastai framework for plant and plant disease classification. Another study [5] on the PlantDoc database utilized an optimal mobile network-based convolutional neural network (OMNCNN) for plant and plant disease classification. It also used the MobileNet model as a feature extraction technique, and the extracted features were classified into respective classes by an extreme learning machine-based classifier. For plant and plant disease classification [6], they applied the PlantDoc dataset and multiple deep learning techniques (MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception). The results, i.e., the probabilities, obtained from each method were then summed using unweighted mean, weighted mean, and unweighted majority methods to obtain the final result. However, 3-ensemble CNNs (3-EnsCNNs) and 5-EnsCNNs exhibited the best performance in plant datasets including and not including leaf diseases, respectively. A trilinear convolutional neural network model (T-CNN) in

plant and plant disease classification [7] method was proposed, and experiments have been performed using PlantVillage [14] and PlantDoc datasets for comparison. In the experiments, they compared and analyzed the pre-trained model refined using the PlantVillage dataset, the pre-trained model with ImageNet, the pre-trained model refined using the PlantDoc dataset, and the pre-trained model refined using both PlantVillage and PlantDoc datasets. In [8], images were acquired using a UAV, a high-resolution (HR) image was created through super-resolution reconstruction (SRR) from the acquired images, and classification was performed. Three experiments for plant disease classification were conducted using HR, low-resolution (LR), and SRR images to compare their performance. Then, the performance of a super-resolution convolutional neural network (SRCNN) was compared with conventional methods. Images restored via SRCNN underwent classification using AlexNet. In [9], SRR was performed; subsequently, plant disease image classification was performed using HR images created using SRR. To perform SRR, a generative adversarial network (GAN) [10] was utilized; in GAN, 23 residual-in-residual dense blocks were used. Furthermore, CNNDiag proposed in a previous study was used as a disease classification method to perform multiclass classification.

As explained in Section 1, motion blur was generated in the image owing to the movement of a camera, which degraded the classification performance; however, all the aforementioned methods did not consider nonlinear motion blur. In a previous study explained in Section 2.2, nonlinear motion blur was considered to improve the plant image segmentation performance.

2.2. Plant Image-Based Semantic Segmentation Method via Motion Deblurring

In motion deblurring-based weed plant image segmentation [15], the wide receptive field attention network (WRA-Net) utilizing the light WRA residual block and upgraded depth-wise separable convolutional blocks was proposed. In addition, the performance of a semantic segmentation model was enhanced through WRA-Net image restoration. In plant image restoration, nonlinear motion blur was considered when performing deblurring. However, previously explained methods only performed plant image-based semantic segmentation, thus presenting a lack of studies on plant image-based classification considering nonlinear motion blur. Thus, this study first proposed and performed plant image classification with nonlinear motion deblurring. The abovementioned previous studies are summarized and compared in Table 1.

Table 1. Summary of existing studies on plant image databases.

Categories	Methods	Advantages	Disadvantages
Motion deblurring and segmentation	WRA-Net [15]	<ul style="list-style-type: none"> - Provides high-quality (HQ) image; - Considers nonlinear motion blur 	<ul style="list-style-type: none"> - Processing time is high; - Does not deal with plant image classification
Classification	XGBoost [1], SVMR [2], AAR [3], DenseNet-121 [4], OMNCNN [5], 3-EnsCNNs and 5-EnsCNNs [6], T-CNN [7]	Processing time is low	Do not consider nonlinear motion blur
SRR and classification	SRCNN and AlexNet [8], GAN and CNN _{Diag} [9]	Provides HR images	<ul style="list-style-type: none"> - Does not consider nonlinear motion blur; - Processing time is high
Nonlinear motion deblurring and classification	PI-NMD and PI-Clas (proposed method)	<ul style="list-style-type: none"> - Provides HQ images; - Considers nonlinear motion blur 	Processing time is high

3. Materials and Methods

3.1. Overall Procedure of the Proposed Method

Flowchart of our method is presented in Figure 1, according to which plant images are used as inputs to the proposed nonlinear motion deblurring network, PI-NMD; the output image is then used as an input to the plant classification network, or PI-Clas, to categorize the data into 28 classes. The structure and detailed description of the proposed PI-NMD and PI-Clas networks are explained in tables and figures in Sections 3.2 and 3.3.

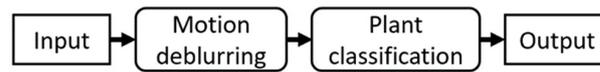


Figure 1. Flowchart of the overall system.

3.2. Detailed Explanation of Structure of PI-NMD Network

The size of both output and input images in the generator is $300 \times 300 \times 3$. Tables 2–4 present the structures of the generator, discriminator, and a residual block in the generator network, respectively. Evidently, the generator and discriminator networks consist of input, (input image), input layer, convolution layer (conv2d), residual block (res_block) [16], additional operation layer (add), upsampling layer (Up2), and a fully connected layer (FC). After conv2d_8 in Table 2 and FC layers in Table 3, tanh and sigmoid activation layers are used, respectively. After the remaining conv2d layers in Tables 2 and 3, a rectified linear unit (ReLU) [17] and leaky ReLU [18] are used, respectively, whereas in Table 4, ReLU is used after conv2d_1, whereas the activation function is used after conv2d_2. Nine res_blocks are used in Table 2. The number of outputs of the FC layer is 1 in Table 3, in which the output is real if the value is closer to 1 and fake if it is closer to 0. Tables 2–4 further present the filter size, stride size, padding size, and filter number of each layer. The connection columns indicate the connection between layers. “#” indicates “number of” in all contents. Parameter numbers of the generator (Table 2) and discriminator (Table 3) are 3,021,638 and 3,369,601, respectively. The structures of the discriminator and generator networks of PI-NMD are depicted in Figure 2. L2–L13 and L2–L8 in Figure 2 indicate the layer number in Tables 2 and 3, respectively. In addition, a single FC layer is used in the discriminator network because the main part of the network comprises convolution layers which extract various features from the input image. The extracted features at the last convolution layer are then used in the FC layer to convert them into a value ranging from 0 to 1 by using a sigmoid activation function.

Table 2. Description of the generator network of PI-NMD.

Layer#	Layer Type	#Filter	Filter Size	Stride Size	Padding Size	Connection
1	input_layer	0	0	0	0	input
2	conv2d_1	64	7×7	1	3	input layer
3	conv2d_2	128	3×3	2	1	conv2d_1
4	conv2d_3	128	3×3	2	1	conv2d_2
5	res_block \times 9	128	3×3	1	1	conv2d_3
6	Up2_1	0	0	0	0	res_block \times 9
7	conv2d_4	64	3×3	1	1	Up2_1
8	Up2_2	0	0	0	0	conv2d_4
9	conv2d_5	64	3×3	1	1	Up2_2
10	conv2d_6	3	7×7	1	3	conv2d_5
11	add	0	0	0	0	conv2d_6 & input_layer
12	conv2d_7	64	4×4	1	1	add
13	conv2d_8	3	3×3	1	1	conv2d_7

Total number of parameters: 3,021,638
 The number of trainable parameters: 3,017,030
 The number of non-trainable parameters: 4608

Table 3. Description of the discriminator network of PI-NMD.

Layer#	Layer Type	#Filter	Filter Size	Stride Size	Padding Size	Connection
1	input_layer	0	0	0	0	input
2	conv2d_1	64	4 × 4	2	1	input_layer
3	conv2d_2	64	4 × 4	2	1	conv2d_1
4	conv2d_3	64	4 × 4	2	1	conv2d_2
5	conv2d_4	64	4 × 4	2	1	conv2d_3
6	conv2d_5	64	4 × 4	1	1	conv2d_4
7	conv2d_6	64	4 × 4	1	1	conv2d_5
8	FC	1	0	0	0	conv2d_6

Total number of parameters: 3,369,601
 The number of trainable parameters: 3,369,601
 The number of non-trainable parameters: 0

Table 4. Description of the residual block.

Layer#	Layer Type	#Filter	Filter Size	Stride Size	Padding Size	Connection
1	input_layer	0	0	0	0	input
2	conv2d_1	128	3 × 3	1	1	input layer
3	conv2d_2	128	3 × 3	1	1	conv2d_1
4	add	0	0	0	0	conv2d_2 & input_layer

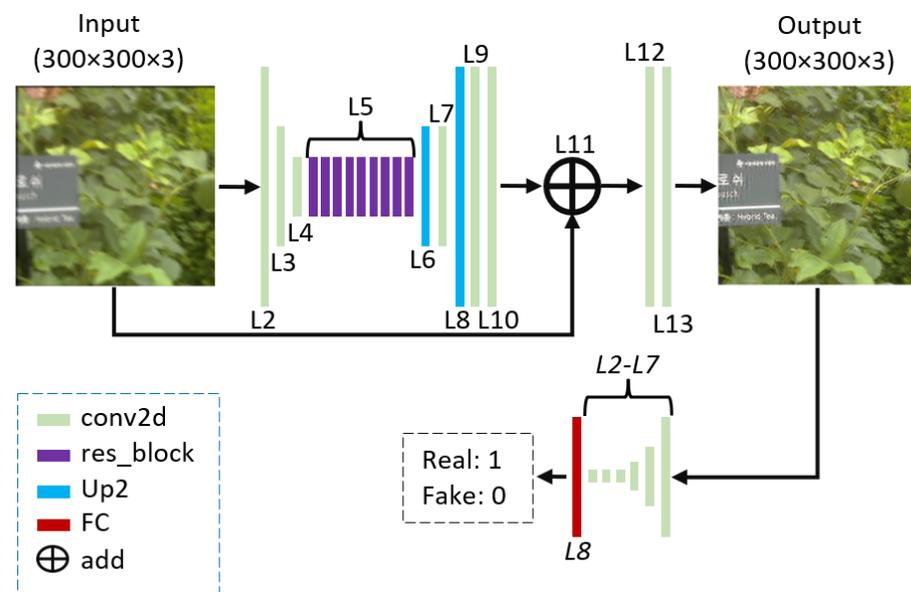


Figure 2. Structure of the PI-NMD.

3.3. Detailed Explanation of Structure of PI-Clas Network

The sizes of output and input images in PI-Clas are 28×1 and $300 \times 300 \times 3$, respectively. Tables 5 and 6 present the proposed PI-Clas network and the residual block used in the PI-Clas network, respectively. As presented in Table 5, conv2d, res_block, FC, add, max pooling (max_pool), and global average pooling layers are used. Furthermore, after the FC layer in Table 5 and the conv2d_1 layer in Table 6, softmax [19] and parametric ReLU [20] are used, respectively. An activation function is not used after the remaining conv2d layers. The output (class #) of FC is 28. The parameter number of PI-Clas is 3,733,532 and that of PI-NMD and PI-Clas combined is 10,124,771. Figure 3 depicts the details of structure of the PI-Clas, where L2–L19 represent layer number in Table 5. In

Table 5, res_blocks at layer# 5, 9, 13, and 17 have different numbers of filters. For example, the res_block at layer# 5 has 64 filters, whereas the res_block at layer# 9 has 128. Therefore, depending on which res_block is used, the number of filters of conv2d_1 in Table 6 is either 64 or 128. Moreover, we used a single FC layer in this network. Here, the extracted features at the last convolution layer are used in the FC layer to convert them into a vector of probabilities by using a softmax function.

Table 5. Description of the proposed PI-Clas.

Layer#	Layer Type	#Filter	Filter Size	Stride Size	Padding Size	Connection
1	input_layer	0	0	0	0	input
2	conv2d_1	64	3 × 3	1	0	input_layer
3	conv2d_2	64	3 × 3	1	0	conv2d_1
4	max_pool_1	0	2	0	0	conv2d_2
5	res_block × 3	64	3 × 3	1	1	max_pool_1
6	conv2d_3	128	3 × 3	1	0	res_block × 3
7	conv2d_4	128	3 × 3	1	0	conv2d_3
8	max_pool_2	0	2	0	0	conv2d_4
9	res_block × 3	128	3 × 3	1	1	max_pool_2
10	conv2d_5	128	3 × 3	1	0	res_block × 3
11	conv2d_6	128	3 × 3	1	0	conv2d_5
12	max_pool_3	0	2	0	0	conv2d_6
13	res_block × 3	128	3 × 3	1	1	max_pool_3
14	conv2d_7	128	3 × 3	1	0	res_block × 3
15	conv2d_8	128	3 × 3	1	0	conv2d_7
16	max_pool_4	0	2	0	0	conv2d_8
17	res_block × 3	128	3 × 3	1	1	max_pool_4
18	GAP	0	0	0	0	res_block × 3
19	FC	class #	0	0	0	GAP

Total number of parameters: 3,733,532

Table 6. Description of the residual block.

Layer#	Layer Type	#Filter	Filter Size	Stride Size	Padding Size	Connection
1	input_layer	0	0	0	0	input
2	conv2d_1	64/128	3 × 3	1	1	input layer
3	conv2d_2	64/128	3 × 3	1	1	conv2d_1
4	add	0	0	0	0	conv2d_2 & input_layer

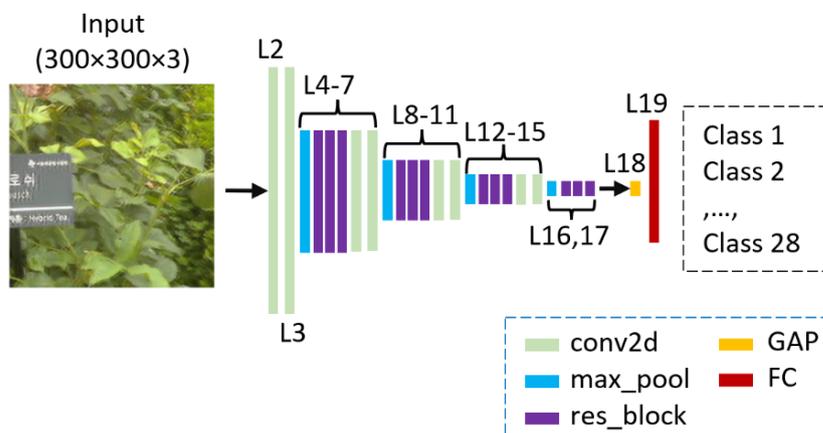


Figure 3. Structure of the PI-Clas.

3.4. Details of the Self-Collected Dataset and Experimental Setup

Here, we conducted experiments using TherVisDb [12], which consists of different types of rose and rose leaf images. This dataset (TherVisDb) was acquired in the summer of 2022. In this study, we used only the visible light images of this dataset and did not use the thermal infrared images. Figure 4 depicts sample images. Table 7 lists image numbers for each class of TherVisDb and the plant names used in the experiment, where “#Image” indicates the number of images, Sets 1 and 2 indicate the split of the dataset for two-fold cross-validation, and the validation set represents the image numbers used in the validation phase. Environmental information at the time of acquiring images, along with the hardware and software information, are as follows: humidity (91%); wind speed (3 m/s); temperature (30 °C); ultra-fine dust (22 $\mu\text{g}/\text{m}^3$); fine dust (24 $\mu\text{g}/\text{m}^3$); ultraviolet index (8); image size (640 \times 512 \times 3); depth (24); image extension (bmp); class number (28); and camera sensor (Logitech C270 [21]). As the images in the dataset were large and a single image contained numerous plants, a cropping operation was performed to create more images of 300 \times 300. Furthermore, overlapping was applied during the cropping operation. Image numbers per class in the dataset were listed in Table 7; the extension of images was changed from “bmp” to “png.” Images in training sets were augmented by using conventional operations (flipped horizontally and rotated three times by 90°). Subsequently, the nonlinear image-blurring method used in [22] was applied to create blurred images in the TherVisDb database. In detail, in the image-blurring method, random trajectories generation was used, in which blur kernels are generated by applying pixel interpolation to vector trajectory. This method can generate motion blur kernels with random trajectories. In other words, this method can simulate more realistic kernels with different levels of nonlinearity. The trajectory generation was made by using the Markov process, which was described as a pseudo code in Algorithm 1 in ref. [22]. In the algorithm, they set the number of iterations, the max length of the movement, and the probability of the impulsive shake to 2000, 60, and 0.001, respectively. In this study, we tuned only two parameters, namely, the number of iterations, and the probability of impulsive shaking. In Section 4.2.1, the experimental results obtained by tuning the parameters are compared.

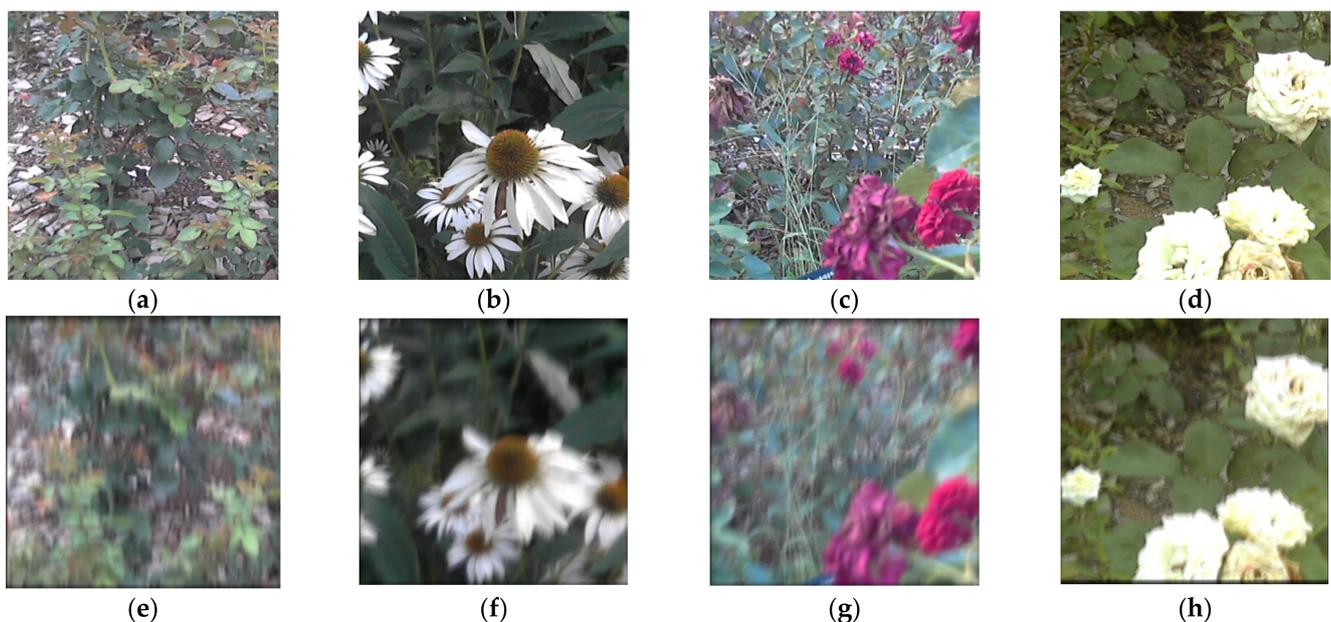


Figure 4. Sample images of TherVisDb dataset: (a) Alexandra; (b) Echinacea Sunset; (c) Rosenau; (d) White Symphonie; (e–h) corresponding blurry images.

Table 7. Description of classes and dataset split.

Class Index	Class Names	#Image	Set 1	Set 2	Validation Set
1	Alexandra	120	54	54	12
2	Belvedere	48	21	21	6
3	Blue river	136	61	61	14
4	Charm of paris	136	61	61	14
5	Cleopatra	152	68	68	16
6	Cocktail	112	50	50	12
7	Duftrausch	176	79	79	18
8	Echinacea sunset	64	28	28	8
9	Eleanor	144	64	64	16
10	Elvis	224	100	100	24
11	Fellowship	208	93	93	22
12	Goldeise	144	64	64	16
13	Goldfassade	184	82	82	20
14	Grand classe	264	118	118	28
15	Just joey	72	32	32	8
16	Kerria japonica	104	46	46	12
17	Margaret	112	50	50	12
18	Oklahoma	312	140	140	32
19	Pink perfume	120	54	54	12
20	Queen elizabeth	120	54	54	12
21	Rose gaujard	312	140	140	32
22	Rosenau	304	136	136	32
23	Roseraie du chatelet	352	158	158	36
24	Spiraea salicifolia l	64	28	28	8
25	Stella de oro	48	21	21	6
26	Twist	288	129	129	30
27	Ulrich brunner fils	120	54	54	12
28	White symphonie	280	126	126	28
Total		4720	2111	2111	498

In addition, the specification of our computer device is as follows: CPU (Intel(R) Core(TM) i7-6700 CPU@3.40 GHz (8 CPUs)); GPU (Nvidia TITAN X (12,233 MB)); and RAM (32,768 MB). The software specifications are as follows: OpenCV (v.4.3.0) [23]; python (v.3.5.4) [24]; Keras API (v. 2.1.6-tf) [25]; and TensorFlow (v.1.9.0) [26].

4. Experimental Results

4.1. Details of Training Setup

For the proposed PI-NMD, the learning rate, epoch number, batch size, optimizer, and loss were set to 0.0001, 150, 8, adaptive moment estimation (ADAM) [27], and binary cross-entropy [28], respectively, whereas for the proposed PI-Clas, they were set to 0.0001, 100, 8, ADAM, and categorical cross-entropy [29], respectively. The validation accuracy curves and training loss curves of PI-NMD and PI-Clas are depicted in Figure 5. The training loss curves of the PI-NMD of each epoch are presented in Figure 5a, whereas the validation loss curves of the PI-NMD of each epoch are presented in Figure 5b. The validation loss curves and training loss curves of the PI-Clas of each epoch are presented in Figure 5c, whereas the validation accuracy curves and training accuracy curves of the PI-Clas of each epoch are depicted in Figure 5d. The training accuracy and loss graphs in Figure 5 indicate that our network was trained sufficiently. In addition, the validation loss and accuracy graphs in Figure 5 indicate that the overfitting did not occur.

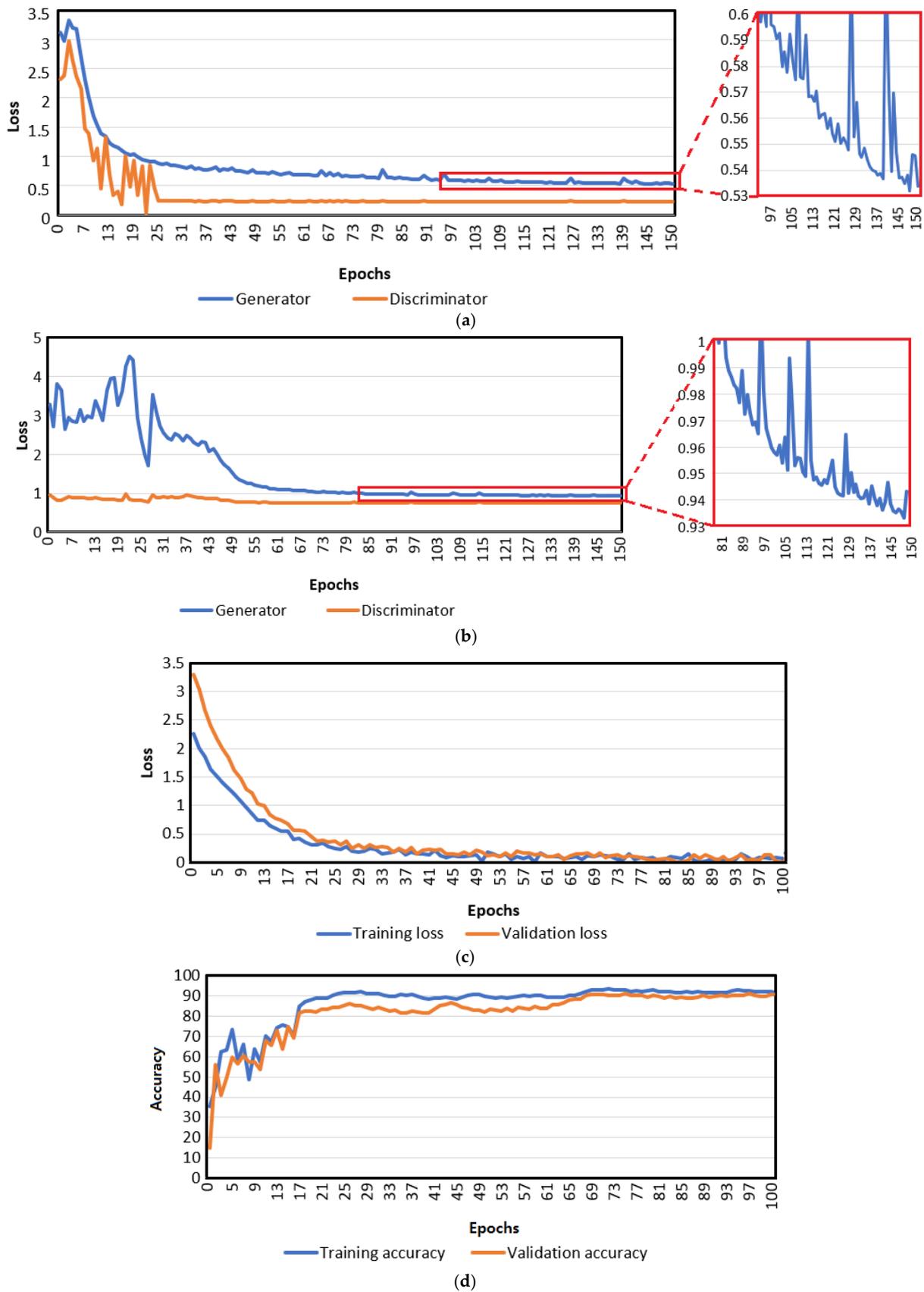


Figure 5. Accuracy and loss curves of the PI-NMD and PI-Clas: (a) training losses of PI-NMD with an enlarged region in a red box; (b) validation losses of PI-NMD with an enlarged region in a red box; (c) validation and training losses of PI-Clas; (d) validation and training accuracies of PI-Clas.

As shown in Figure 5a,b, loss values of the generator network decrease until epoch 150 by very little. For example, in Figure 5a, the loss values of the generator network are 0.56147 and 0.53380 at epoch 115 and 150, respectively. A validation loss value is also decreased at epoch 150 than 115. Therefore, for obtaining the higher accuracy on the image deblurring, a model obtained at epoch 150 was used in this study. In cases of classification (Figure 5c,d), we set the epoch number to 100 as a default number. However, the highest validation accuracy was obtained at epoch 74. Therefore, we used a model obtained at epoch 74.

4.2. Testing with Self-Collected TherVisDb Dataset

4.2.1. Ablation Study

Herein, the following experimental results were obtained from the ablation study. Equations (1) and (2) [30,31] are used to calculate the testing accuracy of PI-NMD. Furthermore, Equations (3)–(5) are used to calculate the testing accuracy of PI-Clas.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{\frac{\left(\sqrt{\sum_{j=1}^M \sum_{i=1}^N (X(i,j) - Y(i,j))^2} \right)^2}{MN}} \right), \tag{1}$$

where X and Y indicate the high-quality image and deblurred image, respectively. Moreover, M and N indicate the width and height of image, respectively.

$$SSIM = \frac{(2\mu_Y\mu_X + C1)(2\sigma_{XY} + C2)}{(\mu_Y^2 + \mu_X^2 + C1)(\sigma_Y^2 + \sigma_X^2 + C2)}, \tag{2}$$

where σ_X indicates standard deviation; μ_X indicates mean value; C1 and C2 are positive constants; and σ_{XY} is a covariance of high-quality image and deblurred image.

$$TPR = (\#TP) / (\#TP + \#FN), \tag{3}$$

$$PPV = (\#TP) / (\#TP + \#FP), \tag{4}$$

where #FP, #TP, #TN, and #FN denote the numbers of false positive, true positive, true negative, and false negative, respectively. These values were used to compute the true F1-score [32], positive predictive values (PPV), and positive rate (TPR), as follows:

$$F1 = 2 \times (PPV \times TPR) / (PPV + TPR). \tag{5}$$

Training and experiments were performed using various PI-NMD structures to design the structure of PI-NMD; the deblurring results of variants of PI-NMD are comparatively presented in Table 8. Method-1 to Method-5 used different numbers of residual blocks. As is evident from Table 8, Method-3 exhibited the highest accuracy. Thus, Method-3 was applied in the next experiment. As is evident from Table 8 and Figure 6, the results of Method-3 were the highest; thus, subsequent experiments were conducted using this structure.

Table 8. Comparison of deblurring accuracies via variants of PI-NMD on a blurred image dataset.

Methods	#res_block	PSNR	SSIM
Method-1	10	20.08	72.90
Method-2	11	19.35	72.81
Method-3 (proposed)	9	21.55	73.10
Method-4	8	18.98	72.76
Method-5	7	18.61	72.71

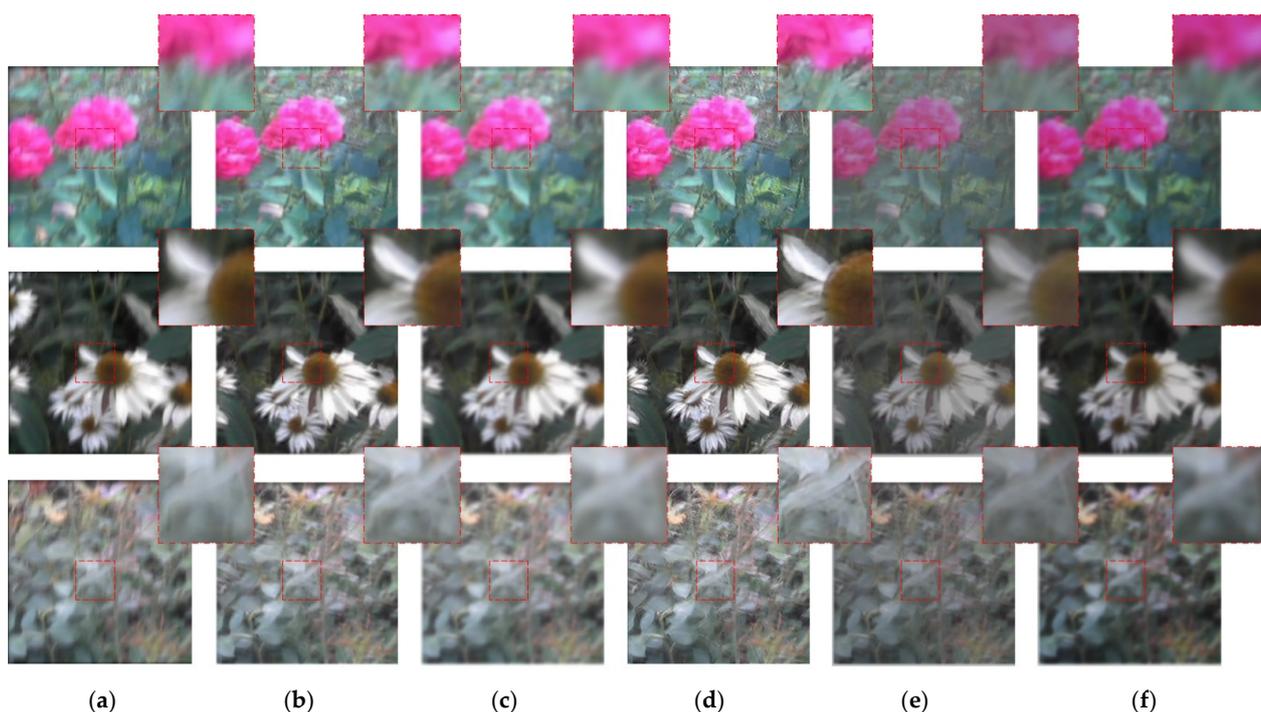


Figure 6. Examples of images deblurred via PI-NMD. From top to bottom, images of Grand Classe, Echinacea Sunset, and Queen Elizabeth: (a) blurry images; deblurred images generated via (b) Method-1, (c) Method-2, (d) Method-3 (proposed), (e) Method-4, and (f) Method-5.

Training and experiments were performed using various PI-Clas structures to design the structure of PI-NMD. The classification results of variants of the PI-Clas are comparatively presented in Tables 9 and 10; in this experiment, an original HQ image dataset was used. Evidently, Method-6 exhibited the highest results. In addition, identical experiments were conducted using a blurred image dataset. As is evident from Tables 11 and 12, Method-6 demonstrated the highest results again; therefore, this structure was applied when conducting subsequent experiments. In subsequent experiments, various blurred datasets were created and used in the deblurring experiment.

Table 9. Comparison of classification accuracies via variants of PI-Clas on an original HQ image dataset.

Methods	TPR	PPV	F1
Method-1	84.73	89.40	87.35
Method-2	87.47	90.66	89.22
Method-3	91.12	92.33	91.71
Method-4	92.03	92.75	92.33
Method-5	89.29	91.50	90.46
Method-6 (proposed)	93.86	93.59	93.58

Table 10. Comparison of classification accuracies via variants of PI-Clas on an original HQ image dataset.

	Top-1	Top-5	Top-10
Method-1	84.73	95.19	97.88
Method-2	87.47	96.46	98.51
Method-3	91.12	98.15	99.36
Method-4	92.03	98.57	99.58
Method-5	89.29	97.30	98.94
Method-6 (proposed)	93.86	99.42	100

Table 11. Comparison of classification accuracies via variants of PI-Clas on a blurred image dataset.

Methods	TPR	PPV	F1
Method-1	77.84	65.18	64.49
Method-2	79.14	67.00	66.97
Method-3	80.86	69.42	70.29
Method-4	81.29	70.03	71.11
Method-5	80.00	68.21	68.63
Method-6 (proposed)	82.15	71.24	72.77

Table 12. Comparison of classification accuracies via variants of PI-Clas on a blurred image dataset.

	Top-1	Top-5	Top-10
Method-1	77.84	85.02	91.64
Method-2	79.14	87.12	92.38
Method-3	80.86	89.92	93.38
Method-4	81.29	90.62	93.62
Method-5	80.00	88.52	92.88
Method-6 (proposed)	82.15	92.02	94.12

Furthermore, Tables 13–15 present the results of conducting experiments using various datasets created by adjusting the parameters (L_{max} and p_b) of the nonlinear image-blurring method explained in a DeblurGAN study [22], where L_{max} and p_b represent the max range of movement and probability of big shake, respectively.

Table 13. Comparison of deblurring accuracies via PI-NMD on variants of the blurred image dataset.

L_{max}/p_b	PSNR	SSIM
60/0.01	16.34	64.81
60/0.005	18.55	68.37
30/0.01	20.76	72.18
30/0.005	21.55	73.10

Table 14. Comparison of classification accuracies via PI-NMD + PI-Clas on blurred datasets blurred with variants of parameters.

L_{max}/p_b	TPR	PPV	F1
60/0.01	83.10	87.97	91.40
60/0.005	85.80	90.24	93.75
30/0.01	88.66	93.12	96.45
30/0.005	90.09	78.64	84.84

Table 15. Comparison of classification accuracies via PI-NMD + PI-Clas on blurred datasets blurred with variants of parameters.

L_{max}/p_b	Top-1	Top-5	Top-10
60/0.01	84.03	72.84	78.98
60/0.005	86.86	75.07	81.01
30/0.01	88.97	77.47	83.36
30/0.005	90.09	94.71	97.60

To summarize the ablation study conducted as set out above, Tables 8–12 present the experimental results of the ablation study conducted to design structures of plant image deblurring and classification methods, respectively. Tables 13–15 present the results of experiments conducted using datasets in which the images were blurred using the two methods.

4.2.2. Comparisons with the Existing Studies

Experiments were conducted to compare our method with the existing motion deblurring and classification methods using the existing open datasets. Section (Comparison with the Existing Plant Image Deblurring Methods) provides the comparison results of previous motion deblurring methods, and Section (Comparison with the Existing Plant Image Classification Methods) provides the comparison results of previous plant classification methods. However, owing to the limited number of studies on plant image-based nonlinear motion deblurring, other similar methods were compared in this section.

Comparison with the Existing Plant Image Deblurring Methods

PI-NMD was compared with other existing image deblurring methods (Blind-DeConV [33], Deblur-NeRF [34], and DeblurGAN [22]), and the accuracies are presented in Tables 16 and 17. In this comparative experiment, all methods were trained by using the self-collected dataset with the same training duration (epochs). Only deblurring accuracies were compared in Table 16, whereas the classification accuracies using the images restored via the deblurring methods are listed in Table 17. The classification was compared using the proposed PI-Clas network. As TPR is identical to the Top-1 accuracy in all cases [35], the Top-1 accuracy was not separately marked.

Table 16. Comparison of motion deblurring accuracies via the PI-NMD and existing methods.

Methods	PSNR	SSIM
Blind-DeConV [33]	17.83	67.87
Deblur-NeRF [34]	20.01	72.15
DeblurGAN [22]	20.89	72.41
PI-NMD (proposed)	21.55	73.10

Table 17. Comparison of classification results via existing deblurring methods and PI-Clas.

Methods	TPR	PPV	F1	Top-5	Top-10
Blind-DeConV [33] + PI-Clas	87.36	72.26	80.34	92.66	95.49
Deblur-NeRF [34] + PI-Clas	88.91	76.30	82.50	93.17	96.25
DeblurGAN [22] + PI-Clas	89.84	77.99	84.15	94.01	96.67
PI-NMD + PI-Clas	90.09	78.64	84.84	94.71	97.60

The images restored using our and existing methods are depicted in Figure 7. As is evident from Tables 16 and 17 and Figure 7, the highest deblurring and classification accuracies were achieved via the proposed method. In other words, previous methods [22,33,34] were used to remove motion blur from blurry images. The output images (deblurred images) of the various methods [22,33,34] were classified via PI-Clas. The classification results were compared in Table 17.

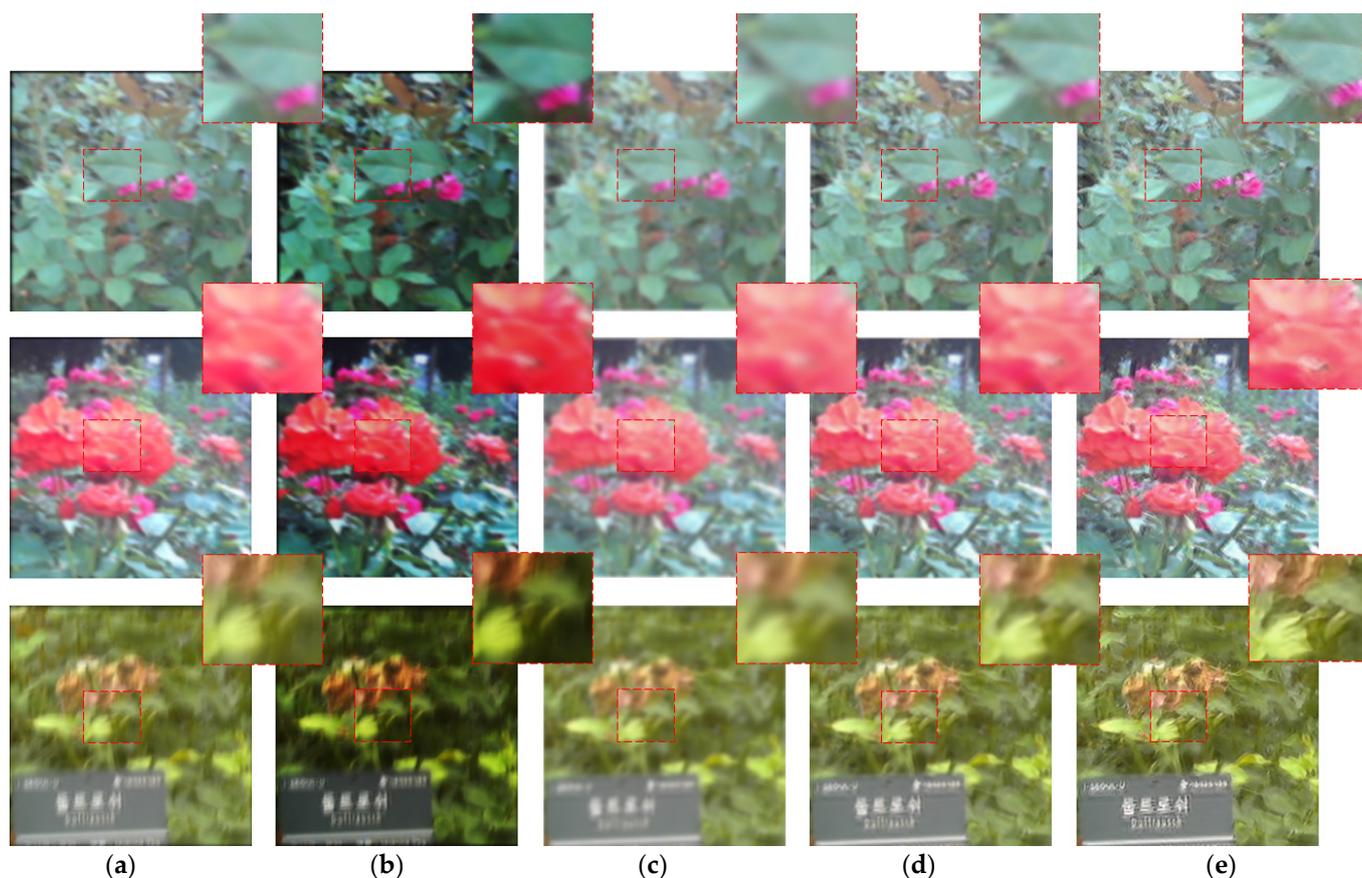


Figure 7. Example of images deblurred via the PI-NMD and existing deblurring methods. From top to bottom, images of Duftrausch, Grand Classe, and Rose Gaujard. (a) Blurry images and deblurred images via (b) Blind-DeConv, (c) Deblur-NeRF, (d) DeblurGAN, and (e) the proposed PI-NMD.

Comparison with the Existing Plant Image Classification Methods

Next, PI-Clas was compared with existing classification methods ((AAR network [3], OMNCNN [5], T-CNN [7], AlexNet [8], and CNNDiag [9]), and the results are comparatively presented in Tables 18–20. Only classification was performed in Tables 18 and 19, whereas deblurring and classification were performed in Table 20. Table 18 presents the accuracies of HQ images, Table 19 shows the accuracies of blurred images, and Table 20 shows the accuracies of deblurred images. Classification performance was compared in all experiments. Evidently, the highest classification accuracy was achieved via the proposed method.

Table 18. Comparison of classification results via existing methods and PI-Clas on the original HQ image dataset.

Methods	TPR	PPV	F1	Top-5	Top-10
AAR network [3]	91.11	88.97	89.29	98.22	100
OMNCNN [5]	92.21	90.82	91.80	98.51	100
T-CNN [7]	93.03	92.20	92.69	98.71	100
AlexNet [8]	91.08	88.53	89.12	98.18	100
CNNDiag [9]	92.15	91.46	91.89	98.68	100
PI-Clas	93.86	93.59	93.58	99.42	100

Table 19. Comparison of classification accuracies via existing methods and PI-Clas on the blurred image dataset.

Methods	TPR	PPV	F1	Top-5	Top-10
AAR network [3]	78.78	64.14	71.84	87.67	90.72
OMNCNN [5]	80.13	66.98	72.21	89.23	92.81
T-CNN [7]	81.14	69.11	72.49	90.74	93.48
AlexNet [8]	78.64	63.05	71.24	87.41	90.35
CNNDiag [9]	81.01	67.22	72.35	90.05	93.15
PI-Clas	82.15	71.24	72.77	92.02	94.12

Table 20. Comparison of classification accuracies via existing methods and PI-Clas on the deblurred image dataset via PI-NMD.

Methods	TPR	PPV	F1	Top-5	Top-10
PI-NMD + AAR network [3]	85.75	70.40	78.22	90.46	94.38
PI-NMD + OMNCNN [5]	88.09	72.87	80.53	93.00	95.83
PI-NMD + T-CNN [7]	89.08	76.88	82.62	93.45	96.60
PI-NMD + AlexNet [8]	84.85	70.38	77.98	90.10	94.18
PI-NMD + CNNDiag [9]	88.43	73.41	81.02	93.41	96.00
PI-NMD + PI-Clas	90.09	78.64	84.84	94.71	97.60

4.3. Testing with Open Datasets

Deblurring and classification experiments were conducted using the existing open datasets PlantDoc [13] and PlantVillage [14]. As our dataset (TherVisDb) does not include plant disease images, only healthy plant images from the two open datasets were used. The number of healthy plant images in PlantDoc was 847 and that of classes was 10. The number of healthy plant images in PlantVillage was 15,084 and that of classes was 12. Our dataset and these two datasets were combined in the training phase, whereas these two datasets were used separately during the testing phase. Figure 8 shows sample images. Table 21 provides the accuracies of using the existing deblurring methods. Table 22 provides the classification accuracies of using the images restored via the existing deblurring methods. Finally, Table 23 presents the classification accuracies of using the images restored via PI-NMD and the existing classification methods. Evidently, the highest deblurring and classification accuracy was achieved via the proposed method.

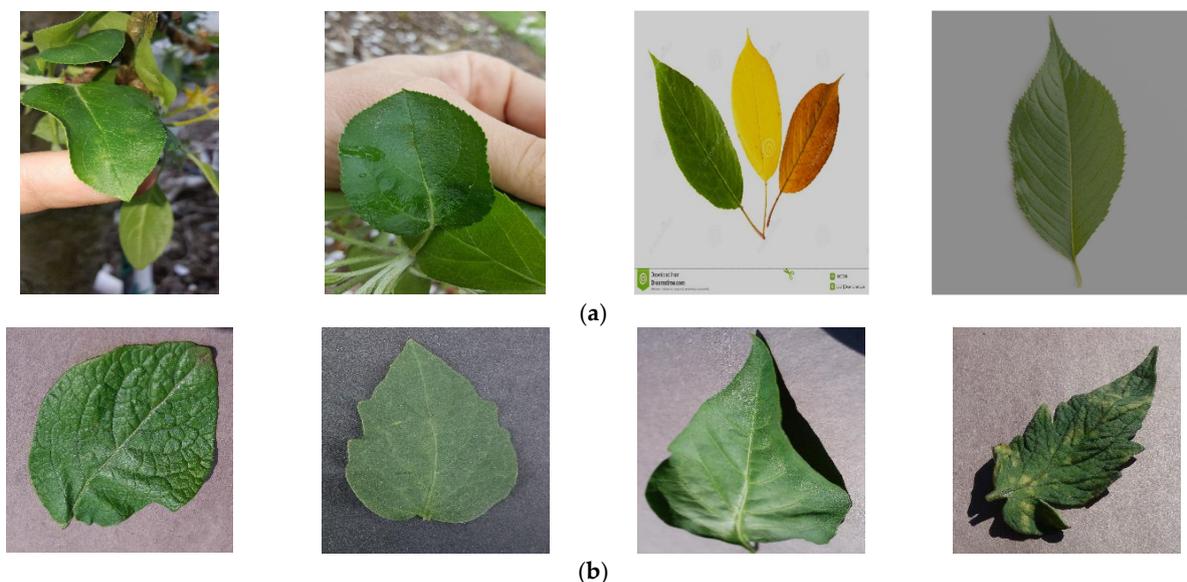
**Figure 8.** Sample images of the open datasets: (a) PlantDoc and (b) PlantVillage.

Table 21. Comparison of motion deblurring accuracies via the PI-NMD and existing methods on the open datasets.

Methods	PlantVillage [14]		PlantDoc [13]	
	PSNR	SSIM	PSNR	SSIM
Blind-DeConV [33]	17.46	67.02	17.56	67.43
Deblur-NeRF [34]	19.50	71.40	19.18	71.37
DeblurGAN [22]	20.63	71.85	20.54	71.52
PI-NMD	20.84	72.96	21.02	72.86

Table 22. Comparison of classification accuracies via the PI-Clas on deblurred image datasets via PI-NMD and other methods on the open datasets.

Methods	PlantVillage [14]			PlantDoc [13]		
	TPR	PPV	F1	TPR	PPV	F1
Blind-DeConV [33] + PI-Clas	86.39	71.96	80.21	79.24	64.21	71.52
Deblur-NeRF [34] + PI-Clas	88.21	75.80	82.48	80.78	67.60	73.98
DeblurGAN [22] + PI-Clas	89.30	77.05	83.94	81.64	69.70	75.57
PI-NMD + PI-Clas	89.79	78.48	84.00	82.21	69.78	76.52

Table 23. Comparison of classification results via existing methods and PI-Clas on deblurred image datasets via the PI-NMD on the open datasets.

Methods	PlantVillage [10]			PlantDoc [5]		
	TPR	PPV	F1	TPR	PPV	F1
PI-NMD + AAR network [3]	85.26	69.54	77.78	77.44	61.71	69.29
PI-NMD + OMNCNN [5]	87.46	72.24	79.74	79.59	64.57	72.06
PI-NMD + T-CNN [7]	88.30	76.30	82.54	80.54	68.25	73.85
PI-NMD + AlexNet [8]	85.10	69.21	77.13	77.12	61.15	69.02
PI-NMD + CNNDiag [9]	88.00	74.45	81.26	80.08	66.42	72.92
PI-NMD + PI-Clas	89.79	78.48	84.00	82.21	69.78	76.52

4.4. Processing Time

The processing times of PI-NMD and PI-Clas are listed in Table 24. Evidently, the frame rate of PI-NMD was 18.6 FPS, whereas that of PI-Clas was 19.37 FPS. The table also presents the giga floating-point operations per second (GFLOPs [36]), parameter numbers, model size, and number of operations of each model.

Table 24. Computation time, GFLOPs, number of parameters, model size, and numbers of multiplication and addition per image of each model.

Model	Processing Time (ms)	GFLOPs	#Parameters (M)	Model Size (MB)	#Multiplication-Addition (Giga)
PI-NMD	55.37	49.1	3.02	12.05	49.22
PI-Clas	51.62	40.8	3.73	44.13	40.78
Total	107.09	89.9	6.75	56.18	90.00

5. Discussion

Nonlinear motion deblurring and classification were performed using plant image datasets in this study. The proposed PI-NMD uses a blurred, low-quality image as input and generates a deblurred, higher-quality image, as shown in Figure 6. As presented in Tables 14 and 15, the proposed PI-Clas method uses the deblurred images of the PI-NMD as input, thus demonstrating higher performance compared with the methods that do not

use the PI-NMD (Tables 10 and 11). Consequently, as explained in Sections 4.2.2 and 4.3, a higher performance compared with that of previous studies was obtained.

Figures 9–11 show error cases of PI-Clas. Images that were improperly restored via PI-NMD resulted in classification errors of PI-Clas. The images contained noises (blue and red pixels), as shown in Figures 9 and 10. The incorrectly classified images during classification generally contained such noises. As shown in Figure 9, noises were observed in all images in the early stages of the training epoch; noise was particularly noticeable in the image of *Spiraea salicifolia* L. (Figure 9d). Further, noise decreases as the number of training epochs increases. Therefore, noise was relatively weak and eventually disappeared in the image of other classes. However, considering the nature of the *Spiraea salicifolia* L. image (Figure 9d), the noise was relatively strong and eventually faded; nevertheless, it did not completely disappear. Hence, the number of training epochs should increase or the hyperparameters and network should be further adjusted during training. The noise in the image on the far right-hand side in Figure 9d is enlarged in Figure 10, and the examples of correctly and incorrectly classified images are shown in Figure 11.

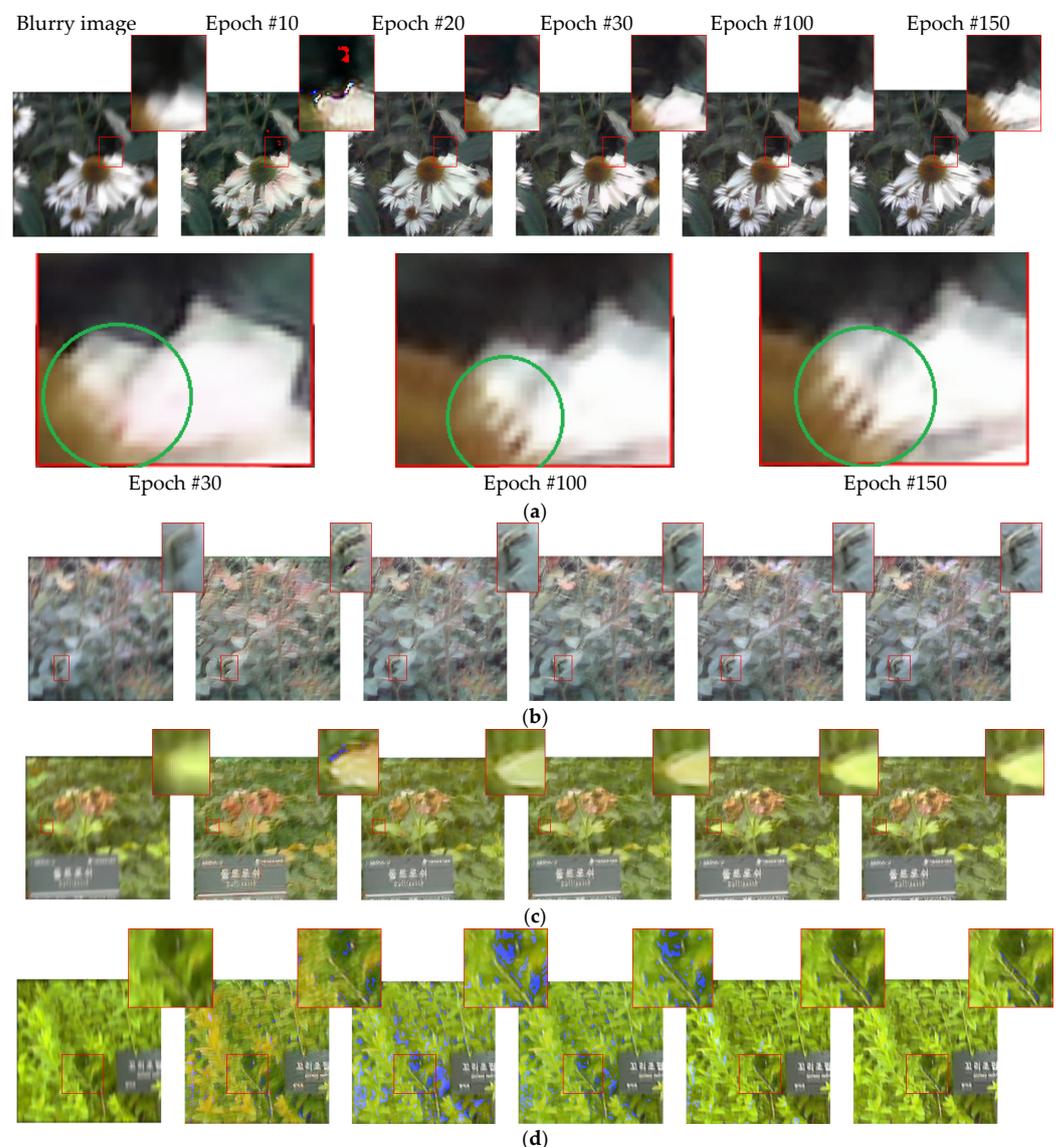


Figure 9. Examples of error cases: (a) Echinacea Sunset; (b) Queen Elizabeth; (c) Rose Gaujard; (d) *Spiraea salicifolia* L. From left to right: blurred images; images deblurred at epoch numbers of 10, 20, 30, 100, and 150.

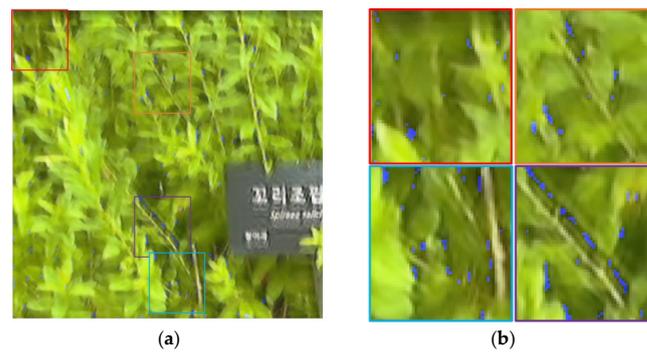


Figure 10. Examples of the error case: (a) deblurred image at epoch number 150 (Figure 9d); (b) regions enlarged from the image (a).

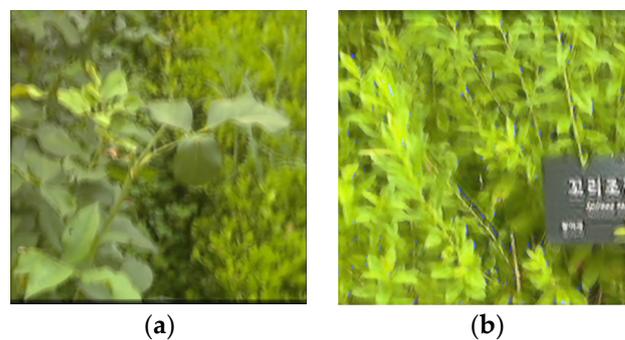


Figure 11. Examples of correctly and incorrectly classified images: (a) Image of Duftrausch correctly classified as Duftrausch image; (b) image of *Spiraea salicifolia* L. (Figure 10) incorrectly classified as Duftrausch image.

The Grad-CAM [37] heatmaps obtained from the final addition layer of the PI-Clas model using original, blurred, and deblurred images are shown in Figures 12–14, respectively. Moreover, the heatmaps are compared in Figure 15. Evidently, the deblurred image-based heatmap obtained from the proposed map was more similar to the original image-based heatmap than to the blurred image-based heatmap depicted in Figure 15.



Figure 12. Example of heatmaps obtained using original images: (a) Blue River; (b,c) Echinacea Sunset; (d) Alexandra.

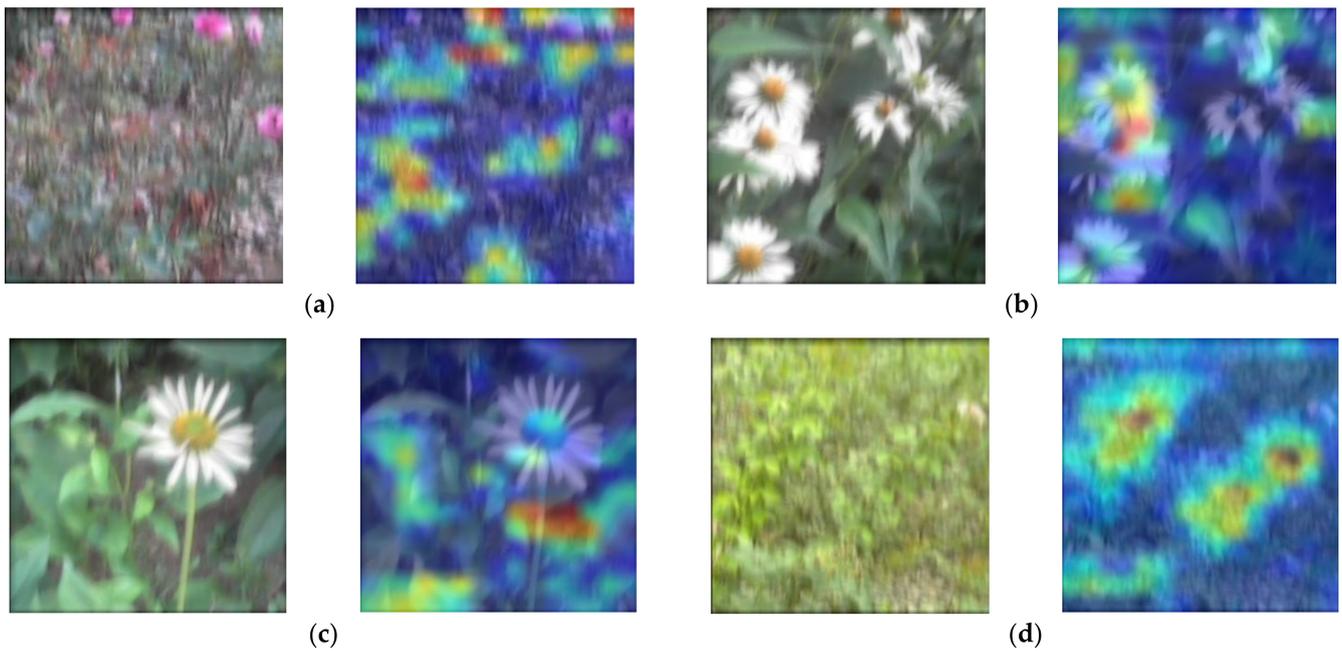


Figure 13. Example of heatmaps obtained using blurred images: (a) Blue River; (b,c) Echinacea Sunset; (d) Alexandra.

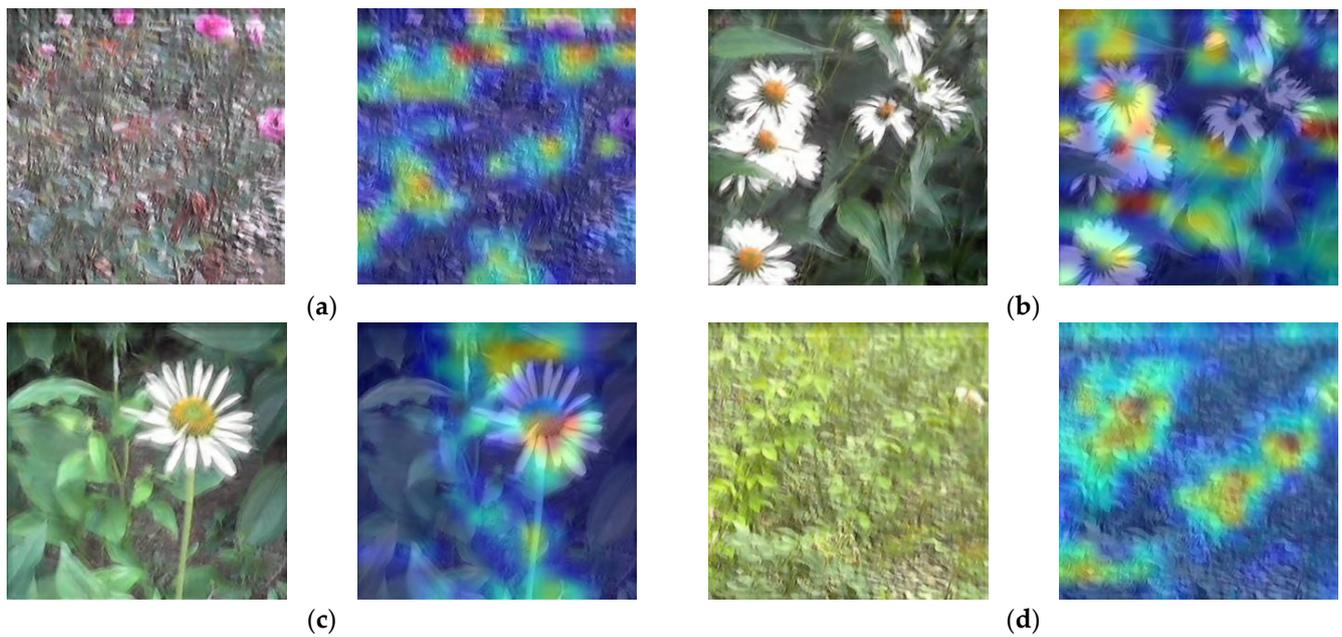


Figure 14. Example of heatmaps obtained using deblurred images: (a) Blue River; (b,c) Echinacea Sunset; (d) Alexandra.

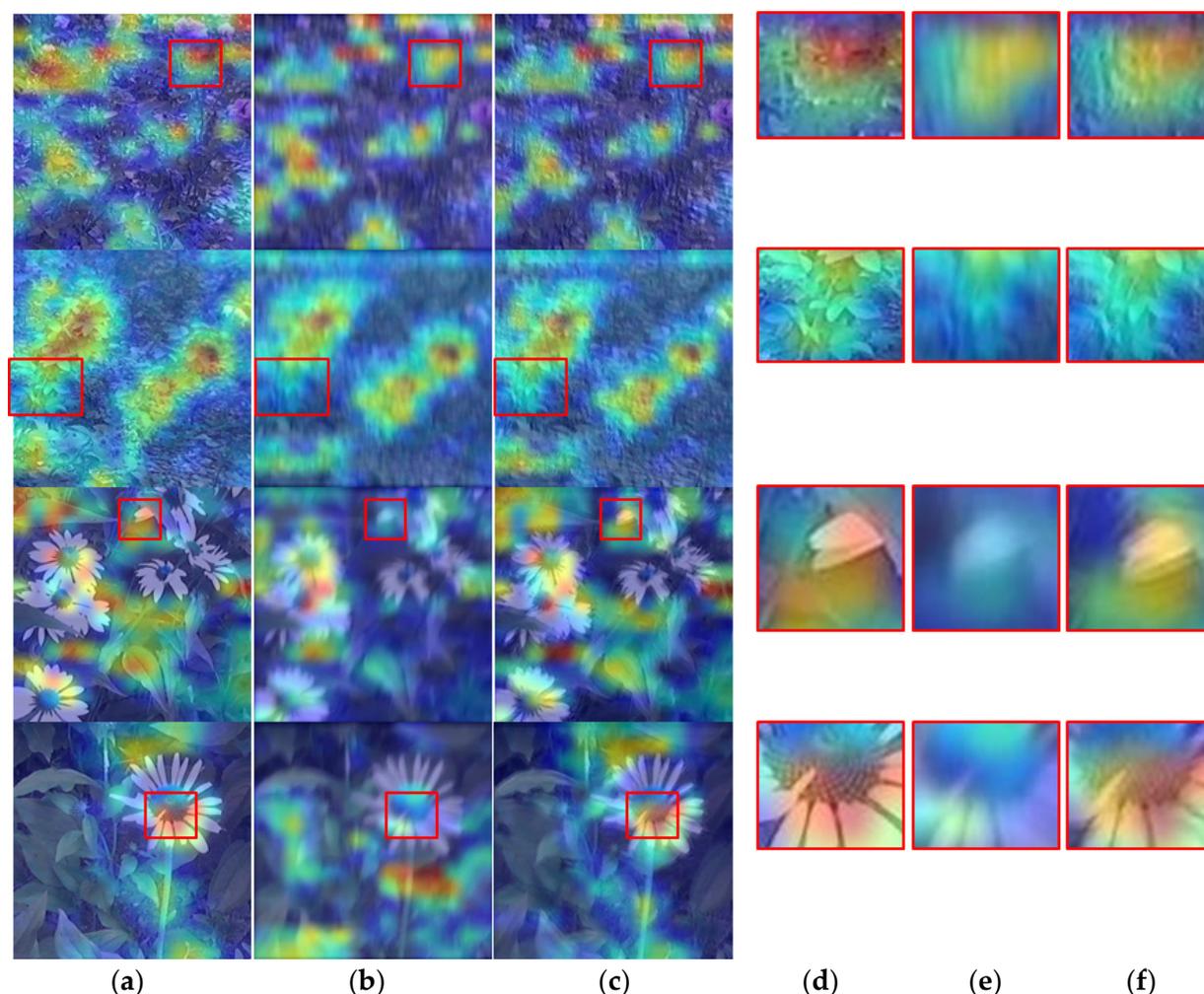


Figure 15. Comparison of the heatmaps shown in Figures 12–14: heatmaps obtained from (a) original images; (b) blurred images; (c) deblurred images. (d–f) are the magnified regions of (a–c), respectively.

6. Conclusions

We proposed nonlinear motion deblurring and classification methods based on plant image datasets and performed relevant experiments using the TherVisDb dataset, which includes different types of rose and rose leaf images. Classification experiments have been performed based on the plant image using the TherVisDb dataset, and our method exhibited higher accuracies of 90.09% (top-1 score) and 84.84% (F1 score) compared with those of existing methods. Furthermore, nonlinear motion deblurring experiments were conducted using plant images in TherVisDb. The proposed deblurring model exhibited higher accuracies of 21.55 (PSNR) and 73.10 (SSIM) compared with those of existing methods. This is because the proposed model was designed based on the ablation study, in which the structure and hyperparameters of the model were tuned based on experimental results employing the dataset with nonlinear motion blur (Tables 8 and 13–15). However, the structures and hyperparameters of the existing methods were not tuned for such datasets in their experiments. Additionally, an experiment was conducted using PlantVillage and PlantDoc open datasets. As presented in Tables 13 and 21, the results obtained using open datasets were lower than those obtained using the self-collected TherVisDb dataset. Moreover, a plant image classification experiment was conducted using PlantVillage and PlantDoc open datasets, and the results obtained using them were lower than those obtained using the TherVisDb dataset. The low accuracy was probably because the PlantDoc open dataset contains a small number of images, most of which include objects (i.e., other than plants). In the case of the PlantVillage open dataset, the accuracy was slightly higher because only

one leaf was included in an image and the background was relatively clean and similar in each. The classification accuracy increased for the images restored via PI-NMD, as can be seen in the experimental results in Tables 17, 20 and 23. As shown in Figures 9–11, errors increased for blue-pixel noise.

In the future, to reduce the classification error observed in Figures 9–11, further research on nonlinear motion deblurring and classification will be performed to improve the performance of the proposed method while considering different deep learning-based image deblurring and classification techniques. Moreover, a study will be conducted using plant thermal images to explore various methods of improving classification performance.

Author Contributions: Methodology, G.B.; validation, J.S.H. and A.W.; supervision, K.R.P.; writing—original draft, G.B.; writing—review and editing, K.R.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT) through the Basic Science Research Program (NRF-2022R1F1A1064291), partly by the NRF funded by the MSIT through the Basic Science Research Program (NRF-2021R1F1A1045587), and partly by the MSIT, Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2020-0-01789) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

Data Availability Statement: Available online: <https://github.com/ganav/PI-Clas/tree/main> (accessed on 18 September 2023).

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

References

1. Wang, J.; Wu, B.; Kohonen, M.V.; Lin, D.; Yang, C.; Wang, X.; Qiang, A.; Liu, W.; Kang, J.; Li, H.; et al. Classification of rice yield using UAV-based hyperspectral imagery and lodging feature. *Plant Phenomics* **2021**, *2021*, 9765952. [[CrossRef](#)] [[PubMed](#)]
2. Bruce, R.W.; Rajcan, I.; Sulik, J. Classification of soybean pubescence from multispectral aerial imagery. *Plant Phenomics* **2021**, *2021*, 9806201. [[CrossRef](#)] [[PubMed](#)]
3. Abawatew, G.Y.; Belay, S.; Gedamu, K.; Assefa, M.; Ayalew, M.; Oluwasanmi, A.; Qin, Z. Attention augmented residual network for tomato disease detection and classification. *Turk. J. Electr. Eng. Comput. Sci.* **2021**, *29*, 2869–2885.
4. Chakraborty, A.; Kumer, D.; Deeba, K. Plant leaf disease recognition using Fastai image classification. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 8–10 April 2021; pp. 1624–1630. [[CrossRef](#)]
5. Ashwinkumar, S.; Rajagopal, S.; Manimaran, V.; Jegajothi, B. Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. *Mater. Today Proc.* **2022**, *51*, 480–487. [[CrossRef](#)]
6. Chompookham, T.; Surinta, O. Ensemble methods with deep convolutional neural networks for plant leaf recognition. *ICIC Express Lett.* **2021**, *15*, 553–565.
7. Wang, D.; Wang, J.; Li, W.; Guan, P. T-CNN: Trilinear convolutional neural networks model for visual detection of plant diseases. *Comput. Electron. Agric.* **2021**, *190*, 106468. [[CrossRef](#)]
8. Yamamoto, K.; Togami, T.; Yamaguchi, N. Super-resolution of plant disease images for the acceleration of image-based phenotyping and vigor diagnosis in agriculture. *Sensors* **2017**, *17*, 2557. [[CrossRef](#)]
9. Cap, Q.H.; Tani, H.; Uga, H.; Kagiwada, S.; Iyatomi, H. Super-resolution for practical automated plant disease diagnosis system. *arXiv* **2019**, arXiv:1911.11341v1.
10. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *arXiv* **2014**, arXiv:1406.2661v1. [[CrossRef](#)]
11. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
12. PI-NMD and PI-Clas Models. Available online: <https://github.com/ganav/PI-Clas/tree/main> (accessed on 18 September 2023).
13. Singh, D.; Jain, N.; Jain, P.; Kayal, P.; Kumawat, S.; Batra, N. PlantDoc: A dataset for visual plant disease detection. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data (CoDS-COMAD), Hyderabad, India, 5–7 January 2020; pp. 249–253. [[CrossRef](#)]
14. PlantVillage Dataset. Available online: <https://www.kaggle.com/datasets/emmarex/plantdisease> (accessed on 16 September 2022).
15. Yun, C.; Kim, Y.W.; Lee, S.J.; Im, S.J.; Park, K.R. WRA-Net: Wide receptive field attention network for motion deblurring in crop and weed image. *Plant Phenomics* **2023**, 1–40, *in press*. [[CrossRef](#)] [[PubMed](#)]

16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]
17. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11–13 April 2011; Proceedings of Machine Learning Research. Volume 15, pp. 315–323. Available online: <https://proceedings.mlr.press/v15/glorot11a.html> (accessed on 18 September 2023).
18. Bing, X.; Naiyan, W.; Tianqi, C.; Mu, L. Empirical evaluation of rectified activations in convolutional network. *arXiv* **2015**, arXiv:1505.00853.
19. Tim, P.; Alexandra, B.; Jun, Z. Understanding softmax confidence and uncertainty. *arXiv* **2021**, arXiv:2106.04972.
20. Kaiming, H.; Xiangyu, Z.; Shaoqing, R.; Jian, S. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv* **2015**, arXiv:1502.01852.
21. Logitech C270 HD Web-Camera. Available online: <https://www.logitech.com/en-us/products/webcams/c270-hd-webcam.960-000694.html> (accessed on 6 March 2023).
22. Kupyn, O.; Budzan, V.; Mykhailych, M.; Mishkin, D.; Matas, J. DeblurGAN: Blind motion deblurring using conditional adversarial networks. *arXiv* **2017**, arXiv:1711.07064.
23. OpenCV. Available online: <http://opencv.org/> (accessed on 6 March 2023).
24. Python. Available online: <https://www.python.org/> (accessed on 6 March 2023).
25. Chollet, F. Keras. California, U.S. Available online: <https://keras.io/> (accessed on 6 March 2023).
26. TensorFlow. Available online: <https://www.tensorflow.org/> (accessed on 6 March 2023).
27. Kingma, D.P.; Ba, J.B. ADAM: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
28. Cross-Entropy Loss. Available online: https://en.wikipedia.org/wiki/Cross_entropy (accessed on 26 October 2022).
29. Categorical Cross-Entropy Loss. Available online: <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy> (accessed on 16 September 2022).
30. Huynh-Thu, Q.; Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.* **2008**, *44*, 800–801. [CrossRef]
31. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
32. Powers, D.M.W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Mach. Learn. Technol.* **2011**, *2*, 37–63.
33. Yang, F.; Huang, Y.; Luo, Y.; Li, L.; Li, H. Robust image restoration for motion blur of image sensors. *Sensors* **2016**, *16*, 845. [CrossRef]
34. Ma, L.; Li, X.; Liao, J.; Zhang, Q.; Wang, X.; Wang, J.; Sander, P.V. Deblur-NeRF: Neural radiance fields from blurry images. *arXiv* **2021**, arXiv:2111.14292.
35. Sawada, A.; Kaneko, E.; Sagi, K. Trade-offs in top-k classification accuracies on losses for deep learning. *arXiv* **2020**, arXiv:2007.15359.
36. Rooks, J.W.; Linderman, R. High performance space computing. In Proceedings of the 2007 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2007; pp. 1–9. [CrossRef]
37. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *arXiv* **2019**, arXiv:1610.02391v4.

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