Deep learning-based multi-task prediction system for plant disease and species detection

Ecological Informatics

Keceli, Ali Seydi; Kaya, Aydin; Catal, Cagatay; Tekinerdogan, Bedir

https://doi.org/10.1016/j.ecoinf.2022.101679

This publication is made publicly available in the institutional repository of Wageningen University and Research, under the terms of article 25fa of the Dutch Copyright Act, also known as the Amendment Taverne. This has been done with explicit consent by the author.

Article 25fa states that the author of a short scientific work funded either wholly or partially by Dutch public funds is entitled to make that work publicly available for no consideration following a reasonable period of time after the work was first published, provided that clear reference is made to the source of the first publication of the work.

This publication is distributed under The Association of Universities in the Netherlands (VSNU) 'Article 25fa implementation' project. In this project research outputs of researchers employed by Dutch Universities that comply with the legal requirements of Article 25fa of the Dutch Copyright Act are distributed online and free of cost or other barriers in institutional repositories. Research outputs are distributed six months after their first online publication in the original published version and with proper attribution to the source of the original publication.

You are permitted to download and use the publication for personal purposes. All rights remain with the author(s) and / or copyright owner(s) of this work. Any use of the publication or parts of it other than authorised under article 25fa of the Dutch Copyright act is prohibited. Wageningen University & Research and the author(s) of this publication shall not be held responsible or liable for any damages resulting from your (re)use of this publication.

For questions regarding the public availability of this publication please contact openscience.library@wur.nl

Ecological Informatics 69 (2022) 101679

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Deep learning-based multi-task prediction system for plant disease and species detection

Ali Seydi Keceli [a,*], Aydin Kaya [a], Cagatay Catal [b], Bedir Tekinerdogan [c]

[a] Department of Computer Engineering, Hacettepe University, Ankara, Turkey
[b] Department of Computer Science and Engineering, Qatar University, Doha, Qatar
[c] Information Technology Group, Wageningen University & Research, Wageningen, The Netherlands

ARTICLE INFO

Keywords:
Plant classification
Multi-task learning
Transfer learning
Deep neural networks
Convolutional neural networks

ABSTRACT

The manual prediction of plant species and plant diseases is expensive, time-consuming, and requires expertise that is not always available. Automated approaches, including machine learning and deep learning, are increasingly being applied to surmount these challenges. For this, accurate models are needed to provide reliable predictions and guide the decision-making process. So far, these two problems have been addressed separately, and likewise, separate models have been developed for each of these two problems, but considering that plant species and plant disease prediction are often related tasks, they can be considered together. We therefore propose and validate a novel approach based on the multi-task learning strategy, using shared representations between these related tasks, because they perform better than individual models. We apply a multi-input network that uses raw images and transferred deep features extracted from a pre-trained deep model to predict each plant's type and disease. We develop an end-to-end multi-task model that carries out more than one learning task at a time and combines the Convolutional Neural Network (CNN) features and transferred features. We then evaluate this model using public datasets. The results of our experiments demonstrated that this Multi-Input Multi-Task Neural Network model increases efficiency and yields faster learning for similar detection tasks.

1. Introduction

Plant disease detection is crucial for preventing plant plagues and minimizing crop risk (Argüeso et al., 2020). Plant diseases adversely impact the quantity and quality of agricultural products and affect food production safety (Chen et al., 2020a), which can in turn harm the financial aspects of the production in the entire area. This is a critical issue particularly for developing countries that depend on only a few or even a single crop (Arsenovic et al., 2019). Visual inspection of plants by experts in the rural areas of developing countries is still the dominant approach, however, this practice is very expensive for large farms (Bai et al., 2018). Due to a large number of plants and phytopathological problems, diagnosing specific diseases is very difficult for agronomists and plant pathologists, often resulting in incorrect conclusions and treatments (Ferentinos, 2018). Therefore, several automated smart approaches using machine learning and deep learning approaches have been developed for the plant disease detection problem (Albayrak et al., 2021; Argüeso et al., 2020; Chen et al., 2020a; Ferentinos, 2018; Golhani et al., 2018; Hernández and López, 2020; Kar et al., 2021; Tiwari et al., 2021; Yang et al., 2022; Yu et al., 2022).

Another important task for plants is the automated classification of plant species (a.k.a., plant classification). It is estimated that there are over 200,000 flowering plant species on Earth, many of which have yet to be discovered, while some of them have already gone various reasons, such as pollution (Govaerts, 2001; Mora et al., 2011; Scotland and Wortley, 2003). The accurate classification of plants is crucial for ecological monitoring and biodiversity conservation, insofar as it can contribute to weed control, the monitoring of endangered species, and species distribution analysis with respect to climate change (Wäldchen et al., 2018). Furthermore, some plant species are important for chemical and medicine industries because of their uses as raw material in the production of medication (Kaya et al., 2019). Subject matter experts can help with this task using plant species catalogs; however, it is a labor-intensive, expensive, and time-consuming process, which requires expertise that is scarce. As such, many different approaches including machine learning and more recently, deep learning, have been proposed

---

* Corresponding author.
E-mail addresses: aliseydi@cs.hacettepe.edu.tr (A.S. Keceli), aydinkaya@cs.hacettepe.edu.tr (A. Kaya), ccatal@qu.edu.qa (C. Catal), bedir.tekinerdogan@wur.nl (B. Tekinerdogan).

https://doi.org/10.1016/j.ecoinf.2022.101679
Received 10 January 2022; Received in revised form 12 May 2022; Accepted 13 May 2022
Available online 20 May 2022
1574-9541/© 2022 Elsevier B.V. All rights reserved.

and evaluated effectively. In some studies, transfer learning from deep models is applied to recognize plants (Yang et al., 2022). In others, state-of-the-art deep learning methodologies are applied in plant disease detection (Yu et al., 2022).

Although several models have been developed to solve both plant disease detection and species classification problems, machine learning/deep learning models learned these tasks in isolation, which means that a separate model has been developed for each task. However, these two tasks are related and share a common representation that can be used to develop more accurate joint models. In machine learning, this learning approach is called Multi-Task Learning (MTL) which leads to better performance according to the empirical and theoretical analysis and investigations (Lai et al., 2021; Zhang and Yang, 2018). MTL attracted the sustained attention of several machine learning researchers in the last ten years, which led to the development of various applications in health informatics, bioinformatics, computer vision, and natural language processing (NLP) (Zhang and Yang, 2017). To the best of our knowledge, MTL has not yet been extensively evaluated in agriculture, and particularly not for plant disease detection and species classification problems. We reused knowledge obtained earlier from similar research in a recent study to investigate the impact of transfer learning approaches on the plant classification models (Kaya et al., 2019) and demonstrated that transfer learning provides better results compared to end-to-end prediction models for plant classification. However, the use of transfer learning together with MTL has not been studied yet in agriculture, particularly regarding the above-mentioned problems. Therefore, in this paper we identified the following two research questions (RQ):

- RQ-1: To what extent can plant classification and plant disease detection problems benefit from the Multi-Task Learning strategy?
- RQ-2: How can transfer learning approaches be integrated into the Multi-Task Learning models?

To answer these research questions, we developed several MTL models and performed several experiments using Multi-Task CNN and pre-trained networks on public datasets. We propose and validate a novel Multi-Task Learning model using Transfer Learning for solving plant disease detection and plant classification problems.

Before proposing a multi-task network, we experimented with mainstream single task CNN and a multi-layer perception network that operates with features from a pre-trained network. We included the results of these tests and performed additional experiments to form an ablation study. The proposed multi-task networks have three main modules: a CNN module that takes raw images as input, a pre-trained module that extracts deep features from images, and a multi-output classification module. We performed experiments by removing the modules one by one to find out which part of the network contributes to our final model. The detailed results are presented in Section 3.3 Ablation Study, which showed that multi-task learning has a positive effect on disease classification.

The contribution of this study is the application of a novel multi-input multi-task network that uses raw images and transferred deep features extracted from a pre-trained deep model to predict both the type and the disease of each plant. Several experiments were conducted on two datasets to demonstrate the effectiveness of the proposed method. In our model, we have two branches, and one of the branches is a typical CNN. This branch has already learned from the input images. We combine features learned from the CNN branch with the extracted features from a pre-trained model in the next layers, from which we learned that the combination of pre-trained features with the features learned from a traditional CNN yields more successful results than the separate ones with a two-branch model. Especially in the case of a small amount of data, feature extraction from a pre-trained model contributes to classification results.

We performed experiments for multi and single-task learning and observed that the multi-task learning strategy provides superior results compared to the single-task experiments. The proposed method achieved state-of-the-art performance compared to the competitors. During the ablation studies, we confirmed that multi-task learning is superior to single-task learning. This fills a lacuna in existing research, since to the best of our knowledge, there is no multi-input multi-task model on plant disease prediction in the literature.

Section 2 presents the methods and materials; Section 3 shows the experimental results; Section 4 discusses our findings and potential threats to the validity; and Section 5 concludes the study.

2. Methods and materials

In this study, we combine two deep learning strategies, namely deep feature extraction and CNN in a single network architecture in a MTL manner. Although deep learning algorithms have been known for a long time, they could not achieve the desired successful results due to lack of sufficient training data and required computing power. Recently, we have seen the emergence of deep learning models trained with large-scale datasets such as ImageNet, which have made deep learning algorithms more and more popular. The most well-known type of deep neural network is the CNN architecture. A CNN can be used either as an end-to-end classifier or as a feature extractor. We integrate these two approaches for plant disease and plant type prediction tasks.

Various variations of CNN models are widely used, and are considered to be state-of-the-art methods. Studies similar to ours generally work with images and the Plant Village dataset used in this study. Related studies on this topic also use images of leaves (Brahimi et al., 2017; Chen et al., 2020a; Kawasaki et al., 2015; Li et al., 2020) to predict plant diseases, as generally, the leaves of plants are the first source of plant disease identification: the symptoms of most diseases appear first on the leaves (Ebrahimi et al., 2017).

2.1. Multi task learning

Multi-task learning (MTL) is a sub-field of machine learning in which commonalities and differences are exploited across tasks, and multiple learning tasks are solved at the same time. Most of the MTL models are based on multi-task neural network architectures (Baltrušaitis et al., 2018). In multi-task architectures used in computer vision, the network is split into different components (Zhang et al., 2014). In these types of networks, a series of convolutional layers are used as base feature extractors. Extracted features are then distributed among all tasks and allocated to different heads specialized for each task.

We can employ the following well-known MTL architectures: shared trunk, cross-talk, prediction distillation, and task routing. In the shared trunk architecture, commonly used in MTL, there is a single input and multiple output branches for each task (Baltrušaitis et al., 2018; Yang et al., 2020; Zhao et al., 2018). In the cross-talk architecture, there are individual feature extraction networks for each specific task (Misra et al., 2016; Ruder et al., 2019). Information is shared between these networks. By combining the predictions or learned features obtained from the different tasks, we arrive at prediction distillation (Vandenhende et al., 2020). In the task routing architectures, the parameter sharing is done at layer level rather than at the feature layer (Strezoski et al., 2019).Task-specific binary masks are applied to given layers, and parameters assign these masks to task-specific network branches. Our proposed model also employs the shared trunk architecture. As can be seen in Fig. 1, a shared feature extraction branch is formed as a classical CNN, the output of which is concatenated with the pre-trained features and given to multiple output branches for different tasks.

In the proposed model, we first constructed the CNN branch using input, convolution, relu, pooling, dropout, and fully connected layers. The CNN structure and relevant parameters are shown in Table 1. We were inspired by the AlexNet architecture (Krizhevsky et al., 2012) while designing the CNN branch of the network, and ended up choosing

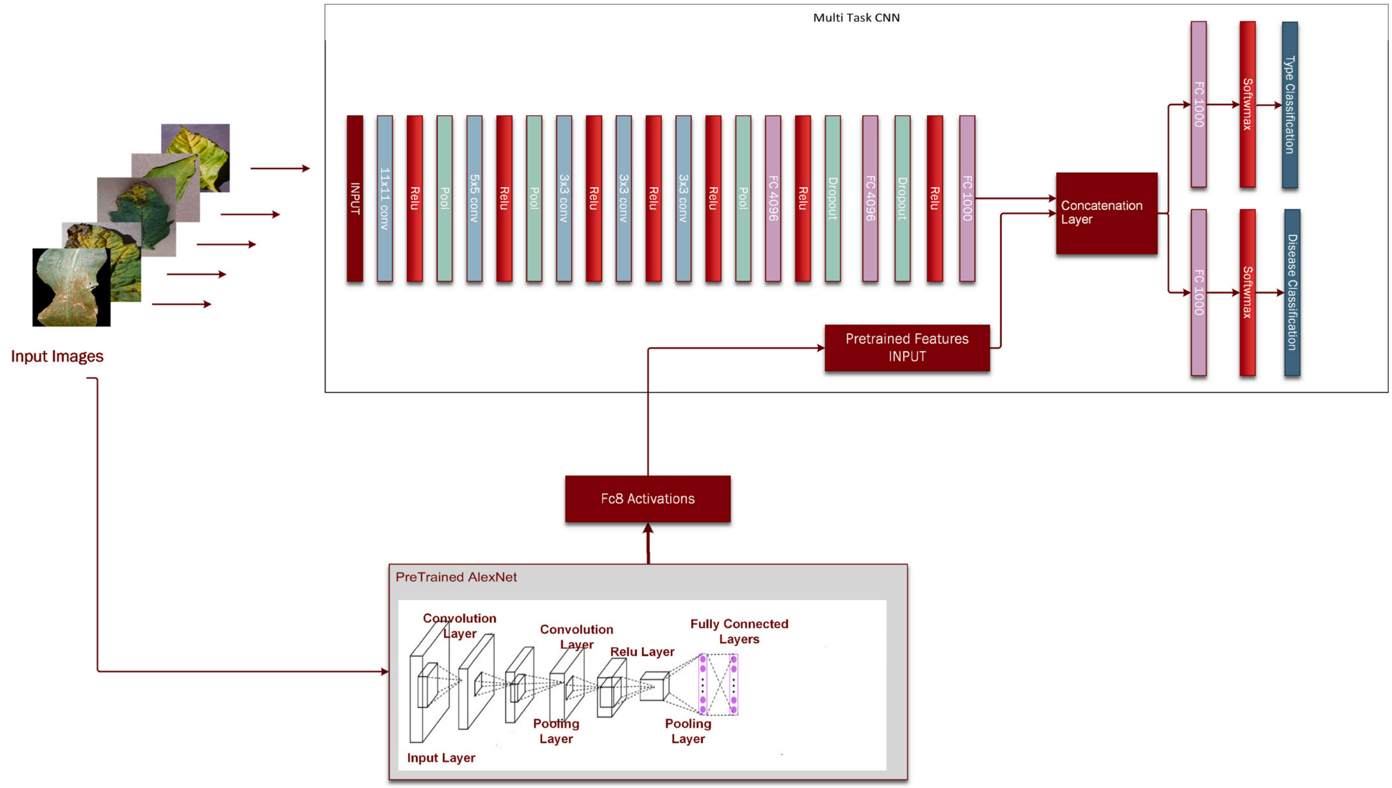


Fig. 1. Architecture of the proposed network.

Table 1
Related parameters of CNN layers.

Layer name	Type	Kernel-Filters	Stride
Input	input	300x300x3	
Conv-1	convolution	11 × 11–48	1
Relu-1	activation		
Norm-1	batch normalization		
Pool-1	max pooling	3 × 3	2
Conv-2	convolution	5 × 5–64	1
Relu-2	activation		
Norm-2	batch normalization		
Pool-2	max pooling	3 × 3	2
Conv-3	convolution	3 × 3–192	1
Relu-3	activation		
Conv-4	convolution	3 × 3–64	1
Relu-4	activation		
Conv-5	convolution	3 × 3–64	1
Relu-5	activation		
Pool-5	max pooling	3 × 3	2
Fc-6	fully connected		
Fc-7	fully connected		
Fc-8	fully connected		

it because of its reduced number of parameters compared with the other pre-trained models. It is still one of the best models for feature extraction compared with the other models (Ahmed and Asif, 2020). AlexNet has fewer parameters and reduces the training time and computational complexity, yet yielded very high accuracy during our experiments.

The entry point of the CNN models is the input layer, which receives raw images or feature vectors as inputs and redirects them to the following convolution layers. The convolution layer is the main layer of the CNN network that performs heavy computations in the network. A large number of 2D filters is convoluted on the images in these layers. The elements in these filters can be considered weights in a traditional artificial neural network model. During the training phase, these filters are updated to extract low and high-level meaningful features from a given image. The pooling layer that follows a convolution layer is applied to reduce the input size and number of parameters. In our architecture, the max pooling operation is applied.

In the max pooling approach, a pooling window is slid over the output of the convolution filters and maximum values inside the sliding window are selected. This process reduces the number of parameters in the network and selects the strong responses from the activation values of convolution layers. In the relu layer, a relu activation function is applied to the output values. The main purpose of applying a relu or other type of activation function is to make the network capable of solving non-linear problems. Although a number of different types of activation functions exist, relu and the hyperbolic tangent functions are the most commonly used ones. Although the sigmoid function was used as the default function in early studies, the current use of this function is very limited due to the vanishing gradient problem, which refers to obtaining very small gradient values (i.e., near to zero) after computing subsequent partial derivatives with the chain rule. Using the relu or weak relu activation functions partly solves this problem.

2.2. CNN model

Another commonly used layer in deep learning is the dropout layer. This layer can drop the network connections randomly using a given probability ratio to prevent the overfitting. An overfit model corresponds too closely to its training data and the learned parameters are highly representative of the training data. As such, this type of model is said to memorize the data instead of the training data and is therefore unable to predict accurately the labels of the unseen data.

Apart from the input, convolutional, pooling, and dropout layers, there are fully connected layers (FC) that consist of different numbers of units. This part of the network is not different than a traditional Artificial Neural Network model. These FC layers represent different layers of abstractions. In many transfer learning studies, the activation values computed by the FC layers are used as features for different machine learning methods. The final layers of CNN models are generally softmax layers that convert the output of the last FC layer into a sum of class probabilities. For this reason, the number of outputs in the last classification layer should be equal to the number of classes for a classification task.

2.3. Proposed model

We developed a multi-input multi-output CNN model from scratch. As seen in Fig. 1, there are two input branches and two output branches. The deep network architecture that can receive more than one type of input at a time is known as a multi-input or multi-stream network. In our two-inputs architecture, a single plant image (i.e., the first input type) and pre-trained features (i.e., the second input type) are combined during the training phase. In the proposed model, we apply transfer learning and extract features from a pre-trained CNN. In machine learning, transfer learning is considered the transfer of knowledge from one learned task to a new task. In the context of neural networks, it entails transferring learned features of a pre-trained network to a new problem. In most of the cases, training a convolutional neural network from scratch in each case is usually not effective when there is an insufficient amount of training data. The common practice in deep learning for such cases is to use a network that is trained for new problems on a large dataset. While we used AlexNet for feature extraction, other well-known pre-trained networks such as VGG16 and VGG19 can also be used for this purpose. The deep features learned from the CNN branch are merged with the features transferred from a pre-trained network and two different loss values are computed for two classification tasks.

The CNN branch extracts high-level features by processing the raw images. The pre-trained features extracted from AlexNet are also introduced to the network as a secondary input to improve the classification results. The Fc-8-layer activation values are extracted and used as input features. During the training phase, pre-trained features for all samples in the dataset are extracted using AlexNet. Both raw images and pre-trained features are used during the training of the final network. After feed forwarding the raw image and the deep features across the network, we merge the deep features and activation values from the last fully connected layer before concatenation. A concatenation layer takes inputs and concatenates them along a specified dimension (the inputs must have the same size in all dimensions except the concatenation dimension). After the concatenation, merged features are forwarded to a fully connected layer including 1000 units. The output of this layer is replicated and moved to two different output branches, having now a fully connected layer, a softmax layer, and a classification layer.

2.4. Training of the model

In a multi-task model, different loss functions are computed and combined. The effect of the individual loss functions to the combined loss can be determined using different weights. Cross entropy loss functions are applied for both species and disease classification. The batch size is determined as 256. For each batch, all instances are feed forwarded across the network and results obtained from each output are stored. These results or labels are used in cross entropy loss computation. These two different loss values are then combined into a single loss value. The cross-entropy losses are shown in Eq. 1 and Eq. 2. Both of these equations are categorical cross-entropy losses, and the combined loss value is the addition of these two values with equal weights. This is shown in Eq. 3. To increase the importance of a prediction task, the weight of this task can be increased. In these equations, log is the natural log, S values are the predicted probabilities, y_i is the class indicator, and θ is the activation function.

$$L_{specie} = -\Sigma_{i=1}^{n} y_i log(S(f_\theta(X_i))) \tag{1}$$

$$L_{disease} = -\Sigma_{j=1}^{m} y_j log(S(f_\theta(X_j))) \tag{2}$$

$$Loss_{combined} = 0.5 {*} L_{specie} + 0.5 {*} L_{disease} \tag{3}$$

After computing the loss value, parameters of the network are updated with gradients using the Adam optimization. Adam is an alternative optimization method that can be applied instead of a classical stochastic gradient decent. It updates the network weights in the training data iteratively. Adam utilizes exponentially smoothed averages (ESA), and the first and second order of the gradient moments. The ESA uses the sample mean. The formulation of Adam is shown in Eq. 4. In the following equation, θ_{t+1} is the new parameter value. m_t and v_t are the first and second moments of the gradient, respectively. The first and the second order moments are not used directly. Smoothing is applied on these values before the update. θ is the learning rate, which defines the update amount of the parameter in each iteration. Finally, ε is a small constant. The formulation of the smoothing is given in Eq. 5 for m_t and v_t.

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \varepsilon} \tag{4}$$

$$m_{tsmooth} = \frac{m_t}{1 - \beta_1^t} \quad v_{tsmooth} = \frac{v_t}{1 - \beta_2^t} \tag{5}$$

The number of the epochs is 1000 and the learning rate is 1e-4. The training parameters of the proposed model (shown in Table 2) are the ones that give the best classification results. After 200 epochs, we did not receive any significant positive update on loss values. On the other hand, we did not want a high epoch number to cause overfitting. We use Adam optimizer because of its adaptive learning rate. Other benefits of the Adam optimizer can be listed as follows: computationally efficient, good for non-stationary objects, works well on problems with noisy or sparse gradients and works well with large datasets. Although we are using the Adam optimizer, we prioritize the model performance and select a relatively small learning rate. A smaller learning rate may allow the model to learn a more optimal or even globally optimal set of weights but it takes time to train. Mini Batch Size is selected as 128. Using a large batch size degrades the quality of the model (Keskar et al., 2016), so we limited the batch size to 128.

2.5. Datasets

We used two publicly available plant datasets for our experiments. The first dataset is the Plant Village dataset (www.plantvillage.org), a publicly available dataset that is widely used in similar studies (Hughes and Salathé, 2015). To predict the performance of our proposed method, our experiments were first conducted on this dataset. The tomato, pepper, potato, and maize datasets of the Plant Village dataset were selected for the training and testing phases. The data were collected under controlled conditions by researchers from the Life Sciences research field. The annotation and disease classification was done by field experts. There were four different species, comprising both healthy samples and those with various diseases – in total, 15 different types of diseases among these species. The diseases together with sample images are listed in Figs. 2, 3, 4, and 5. This first dataset includes 24,488 color images.

The most photographic materials in this Plant Village dataset include images captured in laboratory setups, not in real field settings. To

Table 2
Training parameters.

Parameter	Value
Number of epochs	200
Min batch size	128
Initial learn rate	1.00E-04
Optimizer	Adam

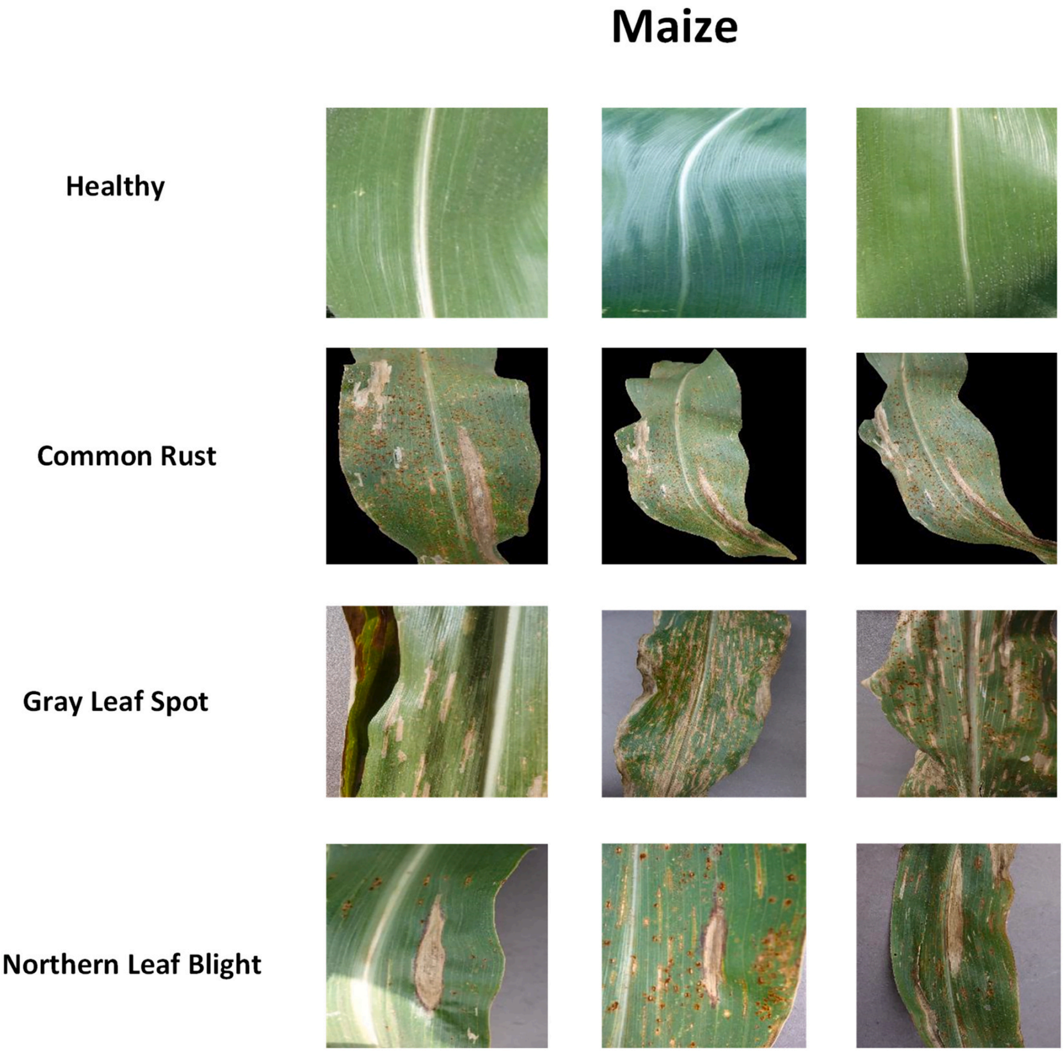


Fig. 2. Healthy and diseased corn images.

increase the generalizability of the proposed method, we use another dataset (FISB) that contains images captured under non-uniform illumination intensities and clutter field background conditions.

The second dataset consists of rice and maize samples collected by Chen et al. (2020a). Some sample images from this dataset are shown in Fig. 6. There are maize and rice samples in this dataset. The samples of the unhealthy plant images are also listed in Fig. 6, and the healthy samples are not included in this dataset. During the experiments, healthy samples are labeled with another class. In this study we propose a CNN-based model; various variations of CNN models are widely used in state-of-the-art research. They generally work with images and the datasets used in this study and other related studies involving images of various plant diseases.

It is important to emphasize here that our proposed models are not dependent on these datasets and can easily be applied to different plant datasets.

3. Experimental results

We conducted three different experiments using two publicly available datasets. The first dataset is the Plant Village dataset (Hughes and Salathé, 2015), and the second one is the data provided by Fujian Institute of Subtropical Botany (FISB dataset), which was used in the paper of Chen et al. (2020a). We implemented multiple input/dual output multi-task CNN for the experiments. All the datasets were split into 70% training, 10% validation, and 20% test sets. The first and the second models are designed for the Plant Village dataset, whereas the third model is designed for the FISB dataset. Classification accuracy, precision, recall, sensitivity, and F1 score metrics were calculated for all the experiments. Table 3 explains these classification metrics and presents the required formula per metric. TP (True Positive) is the count of correctly classified positive class samples; TN (True Negative) is the count of correctly classified negative class samples; FP (False Positive) is the count of negative class samples classified as positive; and FN (False Negative) is the count of positive class samples classified as negative. Confusion matrices are also provided for the experiments. Confusion matrices are visual representations of performances of machine learning models. They are also known as error matrices. Each row of these matrices represents the instances in a predicted class, while each column represents the instances in an actual class (or vice versa). It is the simplest way to see whether the system is mislabeling classes.

3.1. Experiments with Plantvillage dataset

Two experiments were conducted on a subset of the Plant Village dataset, consisting of 24,490 samples in total. We used 4898 test samples from this dataset in our experiments. Our first model classifies the type and the health status of the plant (PlantV-TH). There are four types of plants, namely pepper, potato, tomato, and corn, and only two health statuses (i.e., healthy, unhealthy). Test dataset information of the

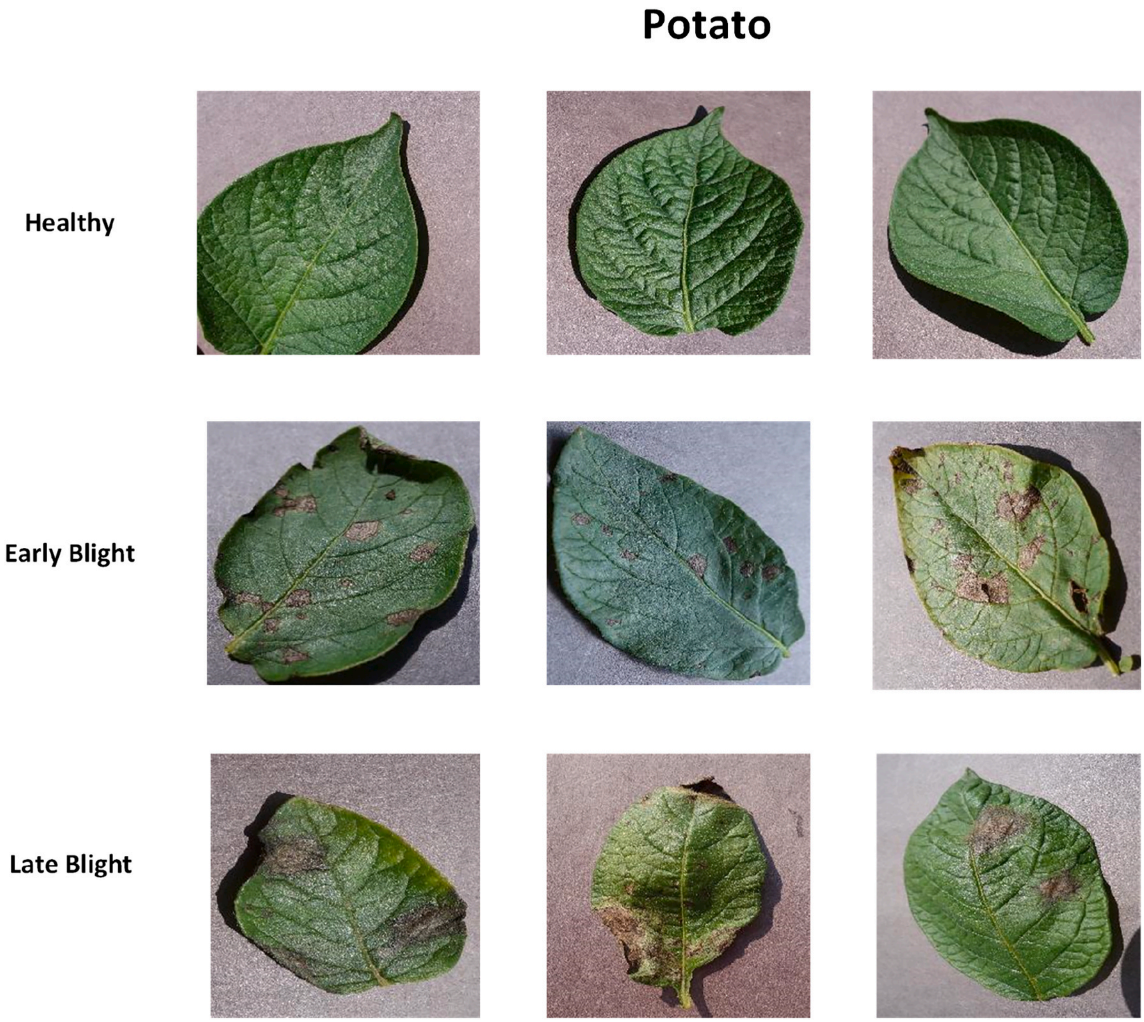


Fig. 3. Healthy and diseased potato images.

experiment is shown in Table 4. Results of the experiments for the PlantV-TH model are presented in Table 5. The confusion matrices of the test results are shown in Figs. 7 and 8. The first model provided 0.98 classification accuracy for both classification branches. Performance on positive classes are almost similar (0.98 and 0.99 precision), but Type Classifier performs better, with 0.99 specificity on samples with negative classes.

The second model classifies the type and the disease status of the plant (PlantV-TD). There are four types of plants, namely pepper, potato, tomato, and corn, and fifteen disease statuses (i.e., healthy; potato: early blight, late blight; tomato: target spot, mosaic virus, yellow leaf curl virus, bacterial spot, early blight, late blight, leaf mold, septoria leaf spot, spider mites; pepper: bacterial spot; gray leaf spot, common rust, northern leaf blight). Test dataset information of the experiment is shown in Table 6. Experimental results for the PlantV-TD model are presented in Table 7. The confusion matrices of the test results are shown in Figs. 9 and 10. The PlantV-TD classifier has a slightly better performance on Type Classification than the PlantV-TH, with 0.99 classification accuracy. However, disease status classifier provides 0.89 classification accuracy. Negative samples classified with high performance (0.99 specificity), and positive samples classified with relatively low performance (0.86 recall). This situation can be explained by the visual similarity among some of the plant diseases (e.g., Tomato target spot and tomato Septoria leaf spot). Disease status classification is a more challenging problem than healthy/unhealthy classification.

3.2. *Experiments with FISB dataset*

A different experiment is conducted on the FISB dataset, which consists of 1160 samples in total. We used 232 test samples from this dataset for our experiments. The model classifies the type and the disease of the plant (FISB-TD). There are two types of plants (i.e., corn and rice), and ten disease types (i.e., Phaeosphaeria spot, corn eyespot, grap leaf spot, southern rusts, Goss' bacterial wilt, rice stackburn, rice leaf smut, rice leaf scald, rice white tip, and bacterial leaf streak). Test dataset information of the experiment is shown in Table 8. Experimental results for the FISB-TD model are presented in Table 9. The confusion matrices of the test results are shown in Figs. 11 and 12. The third model provided 0.97 classification accuracy for plant type classification, and 0.94 classification accuracy for disease type classification. Performance on positive classes and negative classes is the same for plant type classifier (0.97 Recall, and 0.97 specificity). Negative samples classified better for disease type classification (0.99 specificity), and positive samples classified slightly worse than negative ones (0.94 recall).

Comparisons with the other studies in the literature are given for the FISB dataset in Table 10 (Rice) and Table 11 (maize-corn) for disease classification. The comparisons are made for disease prediction, which is less studied compared to species prediction. To the best of our knowledge, there is only one other study that examines this dataset. As can be seen from the results, the multi-task method is overperformed in the other study. Unlike in the Plant Village dataset, the samples of this dataset were collected under complex background conditions.

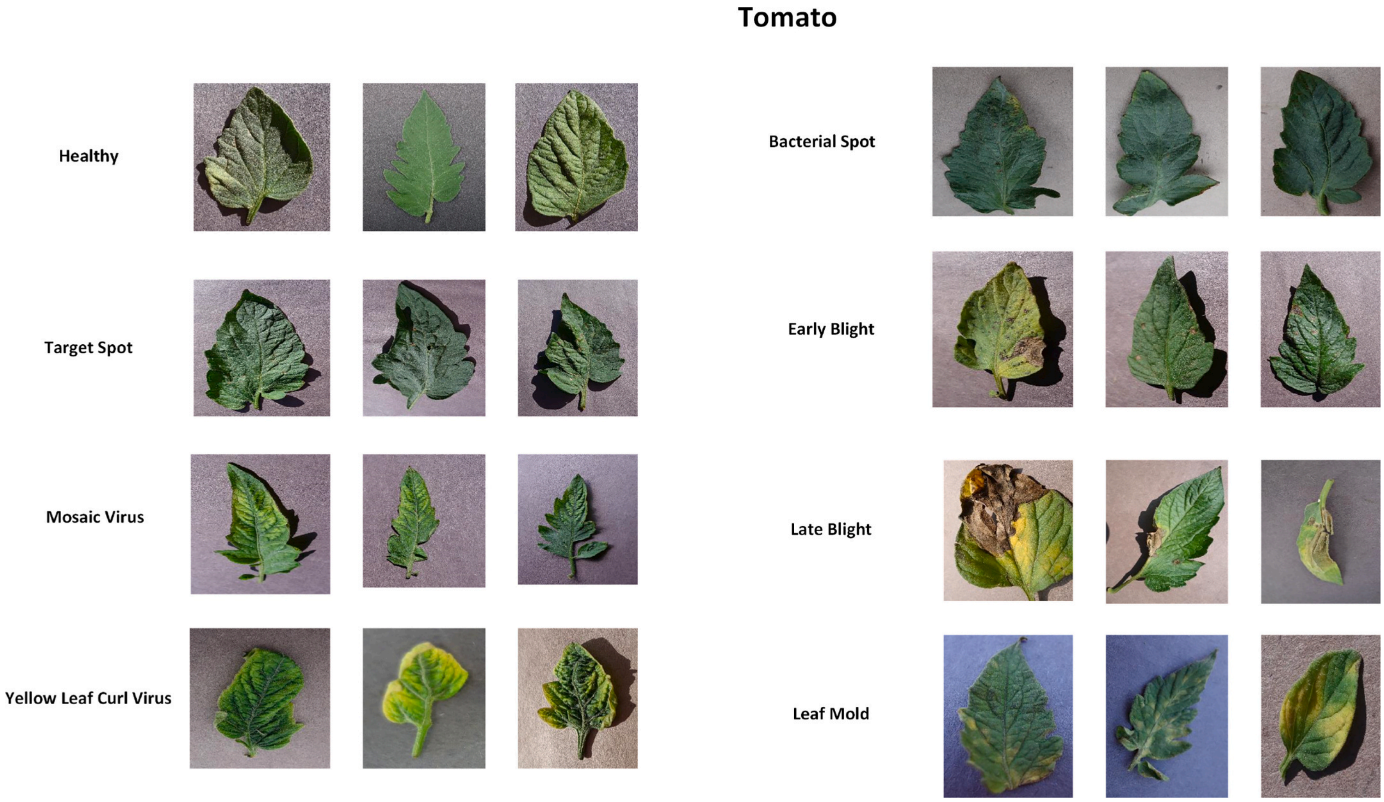


Fig. 4. Healthy and diseased tomato images.

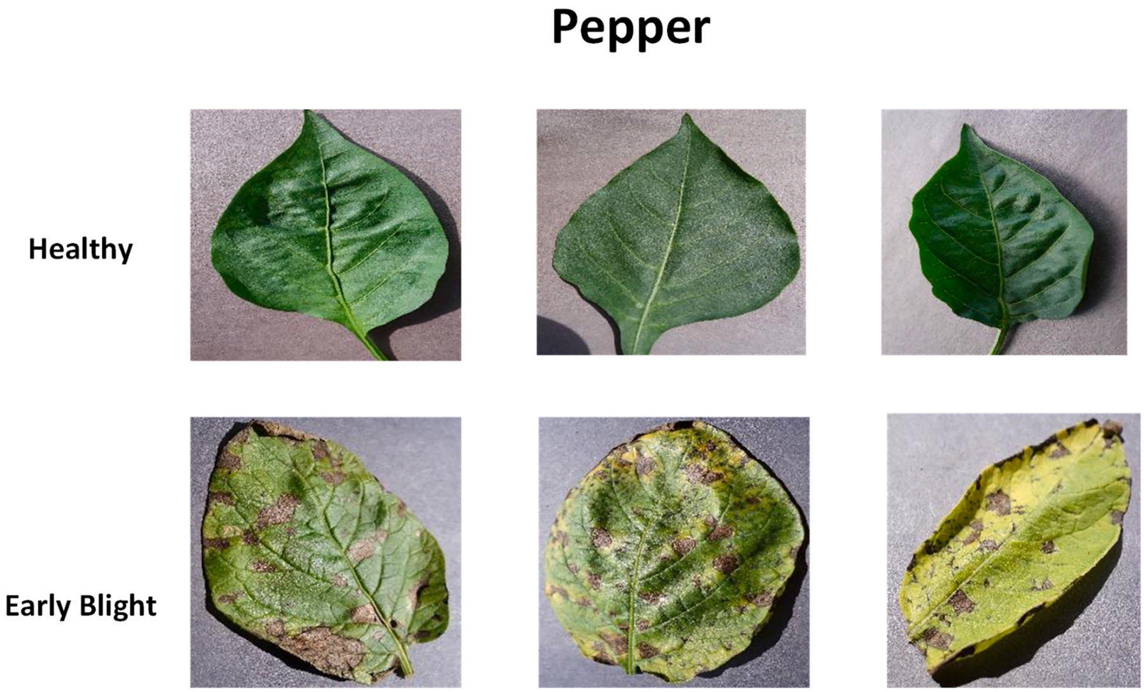


Fig. 5. Healthy and diseased pepper images.

The performance comparisons with the other studies in the literature are shown for the Plant Village dataset in Table 12. Although the Plant Village dataset is publicly available, directly comparing studies' outcomes in the literature is difficult due to varying assumptions and different versions of the dataset. Some of the presented studies are overperformed in our model; these models are trained with data with a limited number of classes and most of them are trained for the diseases of the same species. The superiority of our method can be seen by inspecting the results obtained from one specie. As can be seen from Table 12, our proposed method is superior to (Chen et al., 2020a; Li et al., 2020) which used maize diseases in their experiments. Our models trained with 17 different classes during the experiments. Furthermore,

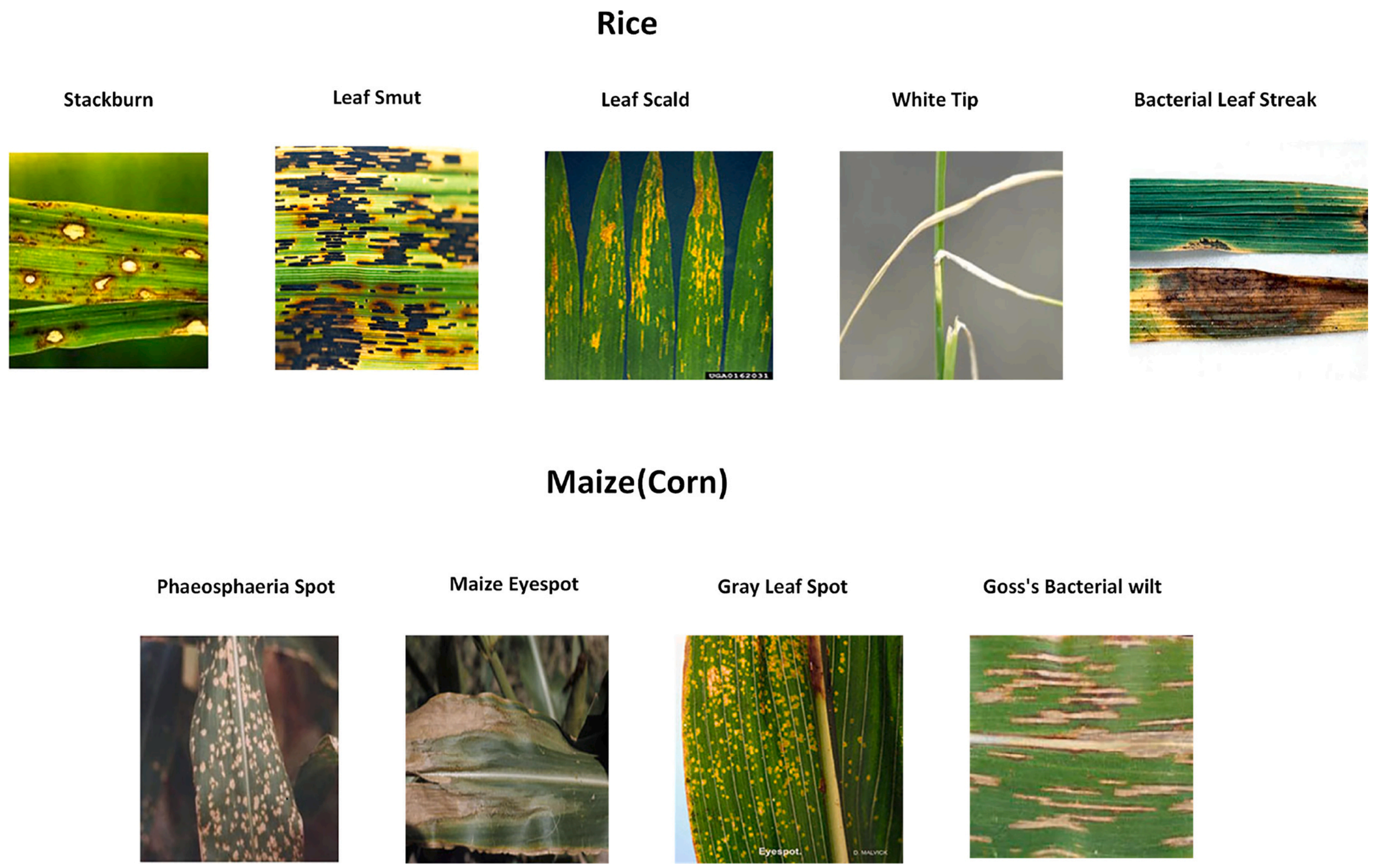


Fig. 6. Healthy and diseased images from 2nd dataset.

Table 3
Classification metrics and the description about these metrics.

Metric name	Description	Formula
Classification Accuracy (CA)	Correct classification ratio of the positive and negative classes.	(TP + TN) / (TP + TN + FN + FP)
Specificity (Spec)	The ratio of True Negatives to all the positives predicted correctly or incorrectly.	TN / (TN + FP)
Recall (Rec)	The ratio of True Positives to all the positives predicted correctly or incorrectly.	TP / (TP + FN)
Precision (Prec)	The ratio of True Positives to all the positives predicted correctly or negatives classified as positive.	TP / (TP + FP)
F1 Score	Harmonic mean of recall and precision.	(2 * Prec * Rec) / (Prec+Rec)

Table 4
Test dataset information for the PlantV-TH model.

Stream	Code	Label name	Count	Percentage
Type	1	Pepper	515	10.42
	2	Potato	399	8.07
	3	Tomato	3240	65.53
	4	Corn	790	15.98
Health status	0	Non-healthy	4030	82.19
	1	Healthy	873	17.81

some studies employ a binary disease classification (Hernández and López, 2020). Among healthy/unhealthy cases, our model has a high and comparable accuracy.

Although there are many studies on species and disease prediction, to the best of our knowledge, there is as yet no other end-to-end model that predicts both at the same time. Chen et al. (2020a)) uses only maize (corn) images and the classification accuracy of our proposed model for only maize diseases are much higher than the results of that study. Training a model with images of various plant species can reduce the average disease classification accuracy.

Table 5
Experimental results for the PlantV-TH model.

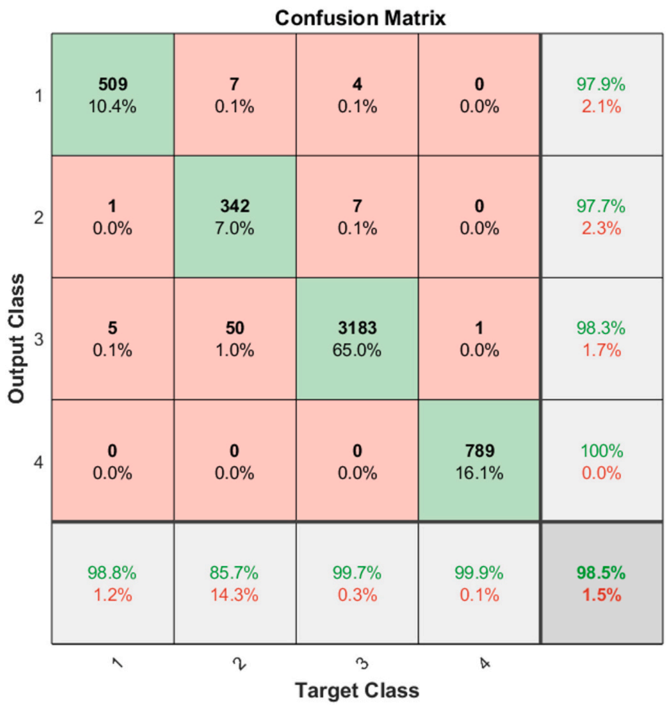
	CA	Spec	Recall	Precision	F1 Score
Type	0.98	0.99	0.96	0.98	0.97
Health status	0.98	0.95	0.99	0.99	0.99

3.3. Ablation study

The multi-input multi-output networks we propose here have three main modules: (a) a CNN module that receives raw images as input, (b) a pre-trained module that extracts deep features from images, (c) and a multi-output classification module that concatenates activations from (a) and (b), and uses a combined loss function to predict about two different classes of the same dataset. For the ablation study, we aim to validate the effectiveness of the proposed models by removing/modifying each module mentioned above.

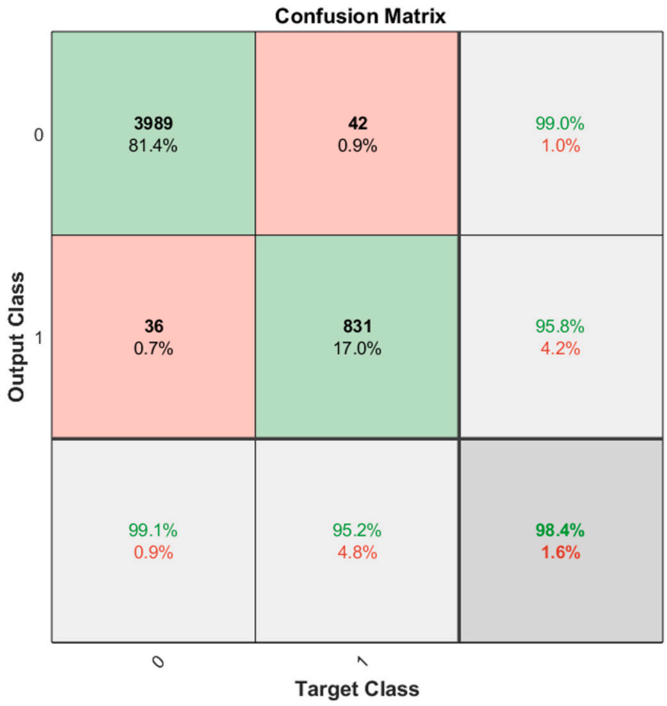
Single Input – CNN – Single Output Networks (SI-CNN-SO): In this model, we removed the pre-trained network branch from the main model, and only classify one label from the datasets. We present the structure of the SI-CNN-SO model in Fig. 13 (a).

Single Input – Pretrained – Single Output Networks (SI-PT-SO): In this model, we removed the CNN branch from the main model, and only classify one label from the datasets. We present the structure of the SI-



Output Class \ Target Class	1	2	3	4	
1	**509** 10.4%	**7** 0.1%	**4** 0.1%	**0** 0.0%	97.9% 2.1%
2	**1** 0.0%	**342** 7.0%	**7** 0.1%	**0** 0.0%	97.7% 2.3%
3	**5** 0.1%	**50** 1.0%	**3183** 65.0%	**1** 0.0%	98.3% 1.7%
4	**0** 0.0%	**0** 0.0%	**0** 0.0%	**789** 16.1%	100% 0.0%
	98.8% 1.2%	85.7% 14.3%	99.7% 0.3%	99.9% 0.1%	**98.5%** **1.5%**

Fig. 7. Confusion matrix of the type output of the PlantV-TH model (Plantvillage dataset).



Output Class \ Target Class	0	1	
0	**3989** 81.4%	**42** 0.9%	99.0% 1.0%
1	**36** 0.7%	**831** 17.0%	95.8% 4.2%
	99.1% 0.9%	95.2% 4.8%	**98.4%** **1.6%**

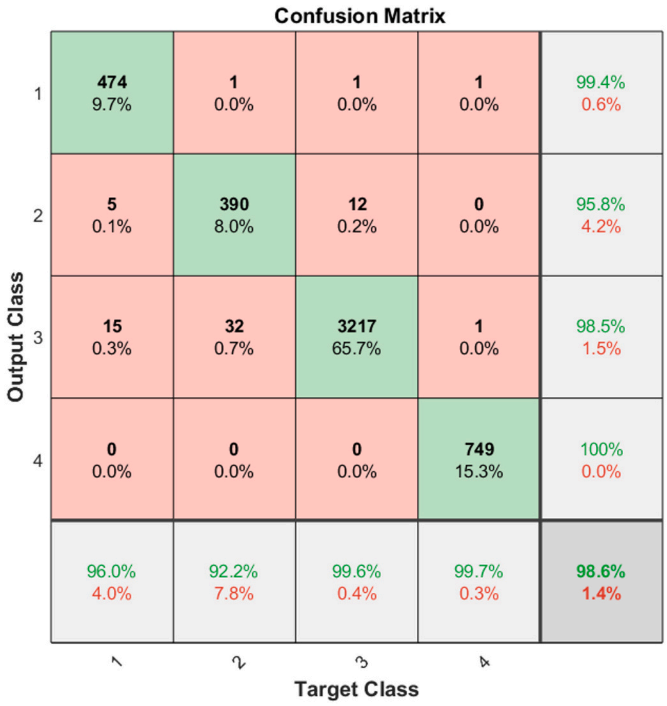
Fig. 8. Confusion matrix of the health status output of the PlantV-TH model (Plantvillage dataset).

Table 6
Test dataset information for the PlantV-TD model.

Stream	Code	Label name	Count	Percentage
Type	1	Pepper	494	10.09
	2	Potato	423	8.64
	3	Tomato	3230	65.95
	4	Corn	751	15.33
Disease status	1	Healthy	886	18.09
	3	PotatoEarly blight	206	4.21
	4	Potato Late blight	189	3.86
	5	Tomato Target Spot	291	5.94
	6	Tomato Mosaic virus	81	1.65
	7	Tomato Yellow Leaf Curl Virus	629	12.84
	8	Tomato Bacterial spot	434	8.86
	9	Tomato Early blight	196	4.00
	10	Tomato Leaf Mold	186	3.80
	11	Tomato Septoria leaf spot	367	7.49
	12	Tomato Two spotted spider mite	347	7.08
	13	Pepper bell Bacterial spot	199	4.06
	14	Corn Gray leaf spot	107	2.18
	15	Corn Common rust	210	4.29
	16	Corn Northern Leaf Blight	205	4.19
	17	Tomato Late Blight	365	7.45

Table 7
Experimental results for the PlantV-TD model.

	CA	Spec	Recall	Precision	F1 Score
Plant type	0.99	0.99	0.97	0.98	0.98
Disease status	0.89	0.99	0.86	0.91	0.87



Output Class \ Target Class	1	2	3	4	
1	**474** 9.7%	**1** 0.0%	**1** 0.0%	**1** 0.0%	99.4% 0.6%
2	**5** 0.1%	**390** 8.0%	**12** 0.2%	**0** 0.0%	95.8% 4.2%
3	**15** 0.3%	**32** 0.7%	**3217** 65.7%	**1** 0.0%	98.5% 1.5%
4	**0** 0.0%	**0** 0.0%	**0** 0.0%	**749** 15.3%	100% 0.0%
	96.0% 4.0%	92.2% 7.8%	99.6% 0.4%	99.7% 0.3%	**98.6%** **1.4%**

Fig. 9. Confusion matrix of the type output of the PlantV-TD model.

PT-SO model in Fig. 13 (b).

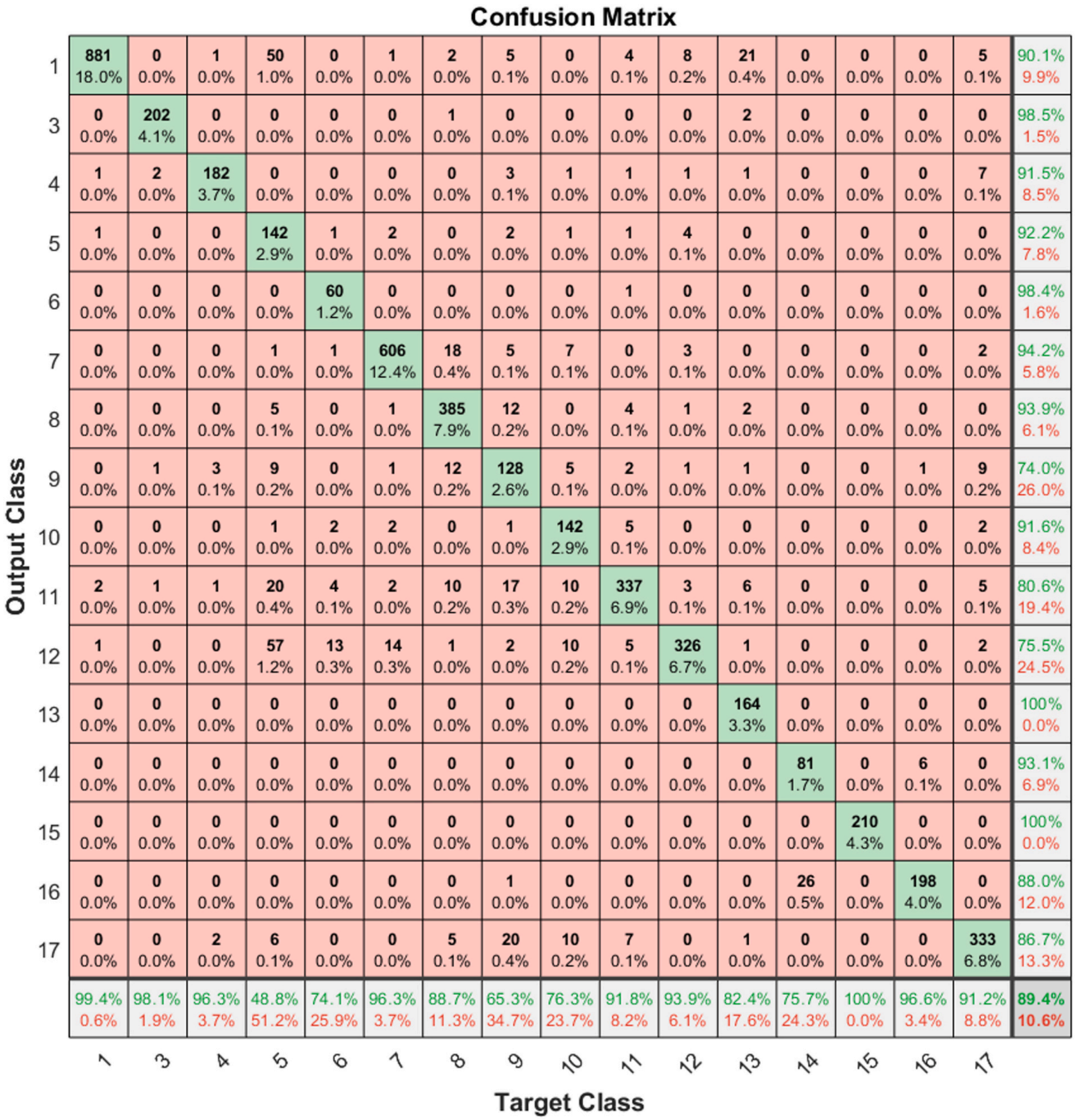
Multi-Input – Single Output Networks (MI-SO): In this model, we keep the CNN and pre-trained branch but only classify one label from the datasets. We present the structure of the MI-SO model in Fig. 13 (c).

Single Input – CNN – Multiple Output Networks (SI-CNN-MO): In this model, we removed the pre-trained network branch from the main model, and classify both of the labels from the datasets. We present the structure of the SI-CNN-MO model in Fig. 13 (d).

Single Input – Pretrained – Multiple Output Networks (SI-PT-MO): In this model, we removed the CNN branch from the main model, and classify both of the labels from the datasets. We present the structure of the SI-PT-MO model in Fig. 13 (e).

The results obtained from the ablation study are presented in Tables 13 and 14.

When we look at the results from the FISB dataset, the proposed



Output \ Target	1	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
1	881 18.0%	0 0.0%	1 0.0%	50 1.0%	0 0.0%	1 0.0%	2 0.0%	5 0.1%	0 0.0%	4 0.1%	8 0.2%	21 0.4%	0 0.0%	0 0.0%	0 0.0%	5 0.1%	90.1% 9.9%
3	0 0.0%	202 4.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.5% 1.5%
4	1 0.0%	2 0.0%	182 3.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.1%	1 0.0%	1 0.0%	1 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	7 0.1%	91.5% 8.5%
5	1 0.0%	0 0.0%	0 0.0%	142 2.9%	1 0.0%	2 0.0%	0 0.0%	2 0.0%	1 0.0%	1 0.0%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.2% 7.8%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	60 1.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.4% 1.6%
7	0 0.0%	0 0.0%	0 0.0%	1 0.0%	1 0.0%	606 12.4%	18 0.4%	5 0.1%	7 0.1%	0 0.0%	3 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	94.2% 5.8%
8	0 0.0%	0 0.0%	0 0.0%	5 0.1%	0 0.0%	1 0.0%	385 7.9%	12 0.2%	0 0.0%	4 0.1%	1 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93.9% 6.1%
9	0 0.0%	1 0.0%	3 0.1%	9 0.2%	0 0.0%	1 0.0%	12 0.2%	128 2.6%	5 0.1%	2 0.0%	1 0.0%	1 0.0%	0 0.0%	0 0.0%	1 0.0%	9 0.2%	74.0% 26.0%
10	0 0.0%	0 0.0%	0 0.0%	1 0.0%	2 0.0%	2 0.0%	0 0.0%	1 0.0%	142 2.9%	5 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	91.6% 8.4%
11	2 0.0%	1 0.0%	1 0.0%	20 0.4%	4 0.1%	2 0.0%	10 0.2%	17 0.3%	10 0.2%	337 6.9%	3 0.1%	6 0.1%	0 0.0%	0 0.0%	0 0.0%	5 0.1%	80.6% 19.4%
12	1 0.0%	0 0.0%	0 0.0%	57 1.2%	13 0.3%	14 0.3%	1 0.0%	2 0.0%	10 0.2%	5 0.1%	326 6.7%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	75.5% 24.5%
13	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	164 3.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
14	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	81 1.7%	0 0.0%	6 0.1%	0 0.0%	93.1% 6.9%
15	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	210 4.3%	0 0.0%	0 0.0%	100% 0.0%
16	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	26 0.5%	0 0.0%	198 4.0%	0 0.0%	88.0% 12.0%
17	0 0.0%	0 0.0%	2 0.0%	6 0.1%	0 0.0%	0 0.0%	5 0.1%	20 0.4%	10 0.2%	7 0.1%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	333 6.8%	86.7% 13.3%
	99.4% 0.6%	98.1% 1.9%	96.3% 3.7%	48.8% 51.2%	74.1% 25.9%	96.3% 3.7%	88.7% 11.3%	65.3% 34.7%	76.3% 23.7%	91.8% 8.2%	93.9% 6.1%	82.4% 17.6%	75.7% 24.3%	100% 0.0%	96.6% 3.4%	91.2% 8.8%	89.4% 10.6%

Fig. 10. Confusion matrix of the disease status output of the PlantV-TD model.

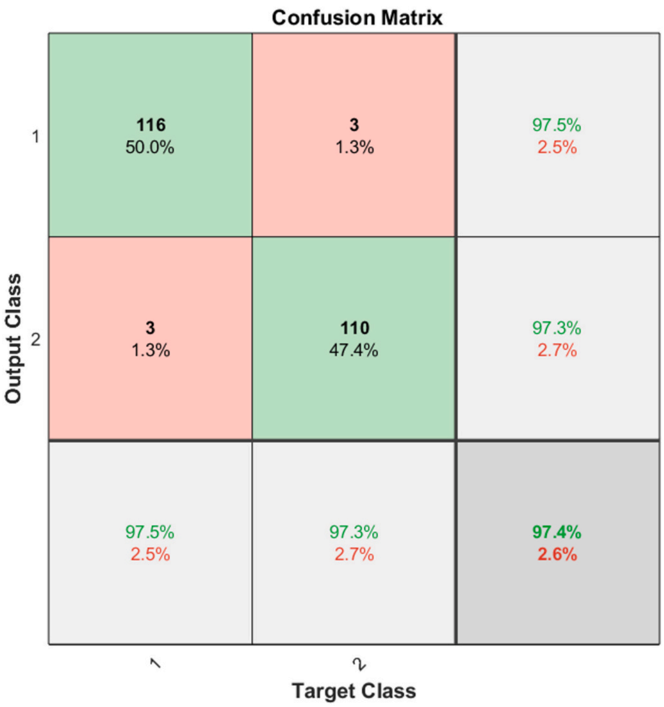
Table 8
Test dataset information for the FISB-TD model.

Stream	Code	Label name	Count	Percentage
Type	1	Corn	119	51.29
	2	Rice	113	48.71
Disease type	3	Phaeosphaeria spot	24	10.34
	4	Corn Eyespot	20	8.62
	5	Gray Leaf Spot	24	10.34
	6	Southern Rusts	28	12.07
	7	Goss's Bacterial wilt	23	9.91
	8	Rice Stackburn	24	10.34
	9	Rice Leaf Smut	22	9.48
	10	Rice Leaf Scald	23	9.91
	11	Rice White Tip	22	9.48
	12	Bacterial Leaf Streak	22	9.48

Table 9
Experimental results for the FISB-TD model.

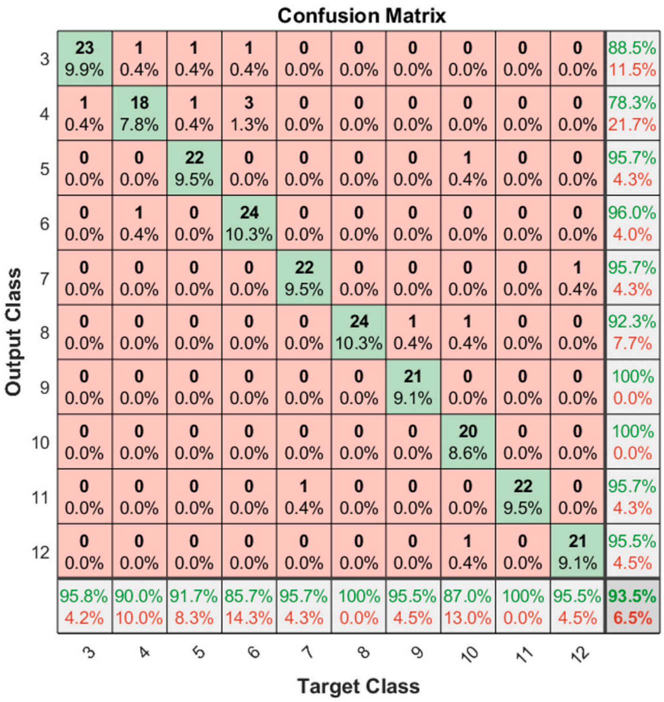
	CA	Spec	Recall	Precision	F1 Score
Plant type	0.97	0.97	0.97	0.97	0.97
Disease type	0.94	0.99	0.94	0.94	0.94

model performed significantly better than the closest performing model by 3% classification accuracy, 5% specificity, 5% precision, and 3% F1-Score for type label. The difference is smaller in disease label by 2% classification accuracy, recall, precision, and F1-Score. MI-SO models provided the best scores between ablation models. In the results from the Plant Village dataset, MI-SO models also provided a better score through the ablation models. The difference is smaller than the results from the FISB dataset, yet the proposed models performed slightly better for the Plant Village dataset as well. The remarkable finding is the models that have a pre-trained branch provide better results on both of the datasets



	1	2	
1	116 50.0%	3 1.3%	97.5% 2.5%
2	3 1.3%	110 47.4%	97.3% 2.7%
	97.5% 2.5%	97.3% 2.7%	97.4% 2.6%

Fig. 11. Confusion matrix of the type output of the third classification model (FISB dataset).



	3	4	5	6	7	8	9	10	11	12	
3	23 9.9%	1 0.4%	1 0.4%	1 0.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	88.5% 11.5%
4	1 0.4%	18 7.8%	1 0.4%	3 1.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	78.3% 21.7%
5	0 0.0%	0 0.0%	22 9.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	0 0.0%	0 0.0%	95.7% 4.3%
6	0 0.0%	1 0.4%	0 0.0%	24 10.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.0% 4.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	22 9.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	95.7% 4.3%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	24 10.3%	1 0.4%	1 0.4%	0 0.0%	0 0.0%	92.3% 7.7%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	21 9.1%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 8.6%	0 0.0%	0 0.0%	100% 0.0%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	0 0.0%	0 0.0%	0 0.0%	22 9.5%	0 0.0%	95.7% 4.3%
12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	0 0.0%	21 9.1%	95.5% 4.5%
	95.8% 4.2%	90.0% 10.0%	91.7% 8.3%	85.7% 14.3%	95.7% 4.3%	100% 0.0%	95.5% 4.5%	87.0% 13.0%	100% 0.0%	95.5% 4.5%	93.5% 6.5%

Fig. 12. Confusion matrix of the disease type output of the FISB-TD model.

Table 10
Comparison of the methods from the literature with FISB Dataset (Rice).

Method	CA	Spec	Recall
Chen et al. (2020a)	0.92	0.95	0.80
Chen et al. (2020b)	0.93	–	–
Proposed	0.97	0.99	0.95

Table 11
Comparison of the methods from the literature with FISB Dataset (Maize).

Method	CA	Spec	Recall
Chen et al. (2020a)	0.80	0.87	0.60
Proposed	0.91	0.88	0.96

Table 12
Comparison of the methods from the literature with Plant Village Dataset.

Method	CA-Disease	Number of classes
Chen et al. (Maize) (Chen et al., 2020a)	0.84	5
Li et al. (Maize) (Li et al., 2020)	0.94	5
Brahimi et al. (Brahimi et al., 2017)	0.96	9
Kawasaki et al. (Kawasaki et al., 2015)	0.94	3
(Lu et al., 2017)	0.95	10
Herdandez et al. (Healty /Non-Healty). (Hernández and López, 2020)	0.96	2
Sharma et al. (Tomato) (Sharma et al., 2020)	0.92	9
Proposed (Healthy/Non-Healthy)	0.98	2
Proposed (Maize Only)	0.96	4
Proposed (Tomato Only)	0.94	9
Proposed	0.89	16

than the single CNN branch models. Plain CNN models performed better on the low-number class samples (type) than on the high-number class samples (disease). We observed that multi-task learning yields superior results compared to a single-task learning. For our case, disease types are also related to the species, such that updating a model with a cost function that combines both species and disease classification will invariably improve performance. Our proposed multi-task models will provide better performance in particular for disease classification.

4. Discussion

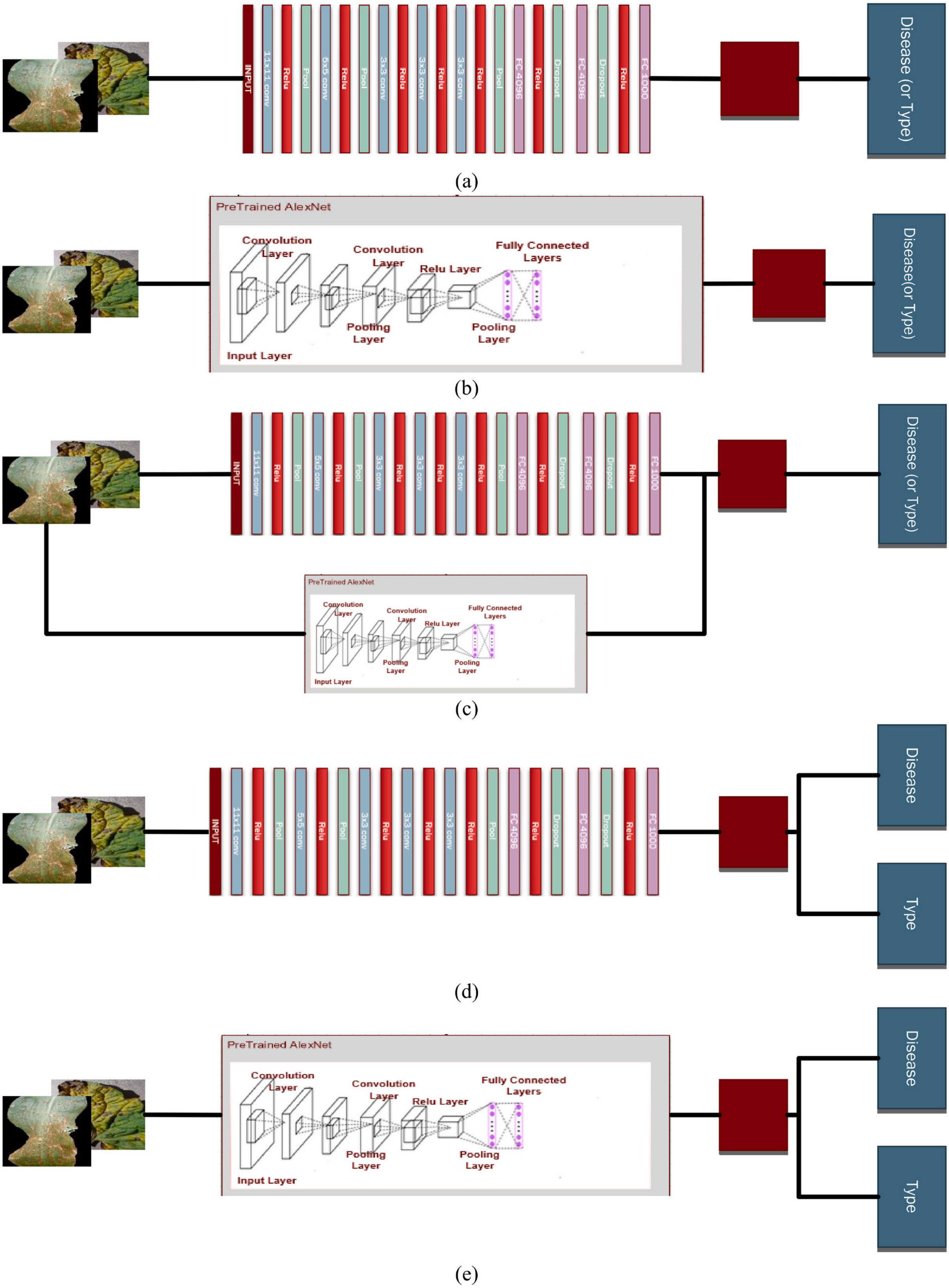
In this study, we proposed and evaluated the performance of several Multi-Input Multi-Task Neural Network models for plant disease detection and species prediction problems. In this section, we discuss our research findings based on the following RQs defined in the Introduction section:

- RQ-1: To what extent can plant classification and plant disease detection problems benefit from the Multi-Task Learning strategy?
- RQ-2: How can transfer learning approaches be integrated into the Multi-Task Learning models?

To respond to these questions, we implemented various models for each dataset. For the first RQ, we demonstrated that our proposed model using Multi-Task Learning can predict both the plant types and disease classes precisely. For the second RQ, we showed that transfer learning can be used together with a separate CNN model to include additional deep features into the Multi-Task Learning-based prediction model. The way we used the transfer learning is focused on the features; however, other researchers might find alternative strategies to benefit from transfer learning for Multi-Task Learning models.

Since this is an experimental study, we should also discuss the potential threats to validity. The validity threats are addressed as follows:

- Internal Validity: We focused on deep learning models instead of traditional machine learning algorithms, because recent studies have proven the superior performance of deep learning models over shallow, learning-based models. We did not apply our model to only one type of problem (e.g., plant type classification and binary health status prediction) but also investigated its performance on the other problem combinations (e.g., plant type classification and disease



(a)

(b)

(c)

(d)

(e)

Fig. 13. Ablation model structures.

Table 13
Ablation model results for FISB dataset.

Model	Label	CA	Spec	Recall	Prec	F1
SI-CNN-SO	Type	0.90	0.90	0.90	0.91	0.90
SI-CNN-SO	Disease	0.57	0.95	0.57	0.61	0.57
SI-PT-SO	Type	0.92	0.90	0.93	0.91	0.92
SI-PT-SO	Disease	0.89	0.99	0.90	0.90	0.90
MI-SO	Type	0.94	0.92	0.96	0.92	0.94
MI-SO	Disease	0.92	0.99	0.92	0.92	0.92
SI-CNN-MO	Type	0.90	0.90	0.89	0.91	0.90
SI-CNN-MO	Disease	0.70	0.97	0.69	0.72	0.70
SI-PT-MO	Type	0.91	0.89	0.93	0.90	0.92
SI-PT-MO	Disease	0.92	0.97	0.92	0.93	0.92
Proposed M.	Type	0.97	0.97	0.97	0.97	0.97
Proposed M.	Disease	0.94	0.99	0.94	0.94	0.94

Table 14
Ablation model results for Plantvillage dataset.

Model	Label	CA	Spec	Recall	Prec	F1
SI-CNN-SO	Type	0.94	0.97	0.88	0.92	0.89
SI-CNN-SO	Disease	0.75	0.98	0.71	0.73	0.72
SI-PT-SO	Type	0.98	0.99	0.98	0.96	0.97
SI-PT-SO	Disease	0.85	0.99	0.83	0.82	0.82
MI-SO	Type	0.98	0.99	0.98	0.96	0.97
MI-SO	Disease	0.86	0.99	0.85	0.84	0.83
SI-CNN-MO	Type	0.95	0.98	0.92	0.92	0.92
SI-CNN-MO	Disease	0.77	0.98	0.72	0.74	0.72
SI-PT-MO	Type	0.97	0.98	0.98	0.96	0.97
SI-PT-MO	Disease	0.87	0.98	0.82	0.86	0.83
Proposed M.	Type	0.99	0.99	0.97	0.98	0.98
Proposed M.	Disease	0.89	0.99	0.80	0.91	0.87

class prediction). The Alexnet pre-trained model that we used can easily be switched with other pre-trained models.

- External Validity: We preferred two widely used datasets for our experiments and considered their sizes before building our models and performing experiments. Our models are not dependent on these datasets; however, the performance on other datasets might be different than the performance values reported here. We consider that acceptable performance can be achieved after adapting the current models to the new datasets.
- Construct Validity: We split the datasets into three categories, namely training, validation, and test, as do most other machine learning researchers. Moreover, we used 1000 epochs for our experiments. This validation strategy was selected in order to avoid randomness in the data. Several well-known evaluation metrics were selected to report the experimental results. Additional experiments can be performed when new datasets are created.

Most of the studies in the literature are based on single input, single output networks (Brahimi et al., 2017; Chen et al., 2020a; Hernández and López, 2020; Kawasaki et al., 2015; Li et al., 2020; Lu et al., 2017; Sharma et al., 2020). In this study, we implement a multi-task network with three main modules: (a) a CNN module that receives raw images as input, (b) a pre-trained module that extracts deep features from images, (c) and a multi-output classification module that concatenates activations from (a) and (b), and uses a combined loss function to make predictions for two different classes of the same dataset (multi-task). To prove the efficiency of our model, we experimented with different single-input single task and multi-task network structures. Results of the ablation studies show that multi-task learning has a positive effect on disease classification. The ablation study results also show that a combination of learned features with transferred features improves the accuracy. Most of the studies in this domain utilize learned features or transferred features only (Brahimi et al., 2017; Chen et al., 2020a; Chen et al., 2020b; Hernández and López, 2020; Kawasaki et al., 2015; Li et al., 2020; Lu et al., 2017; Sharma et al., 2020). The proposed multi-input multi-task method has two branches: one of the branches is a typical CNN, while the other is a regular neural network that receives transferred features as input. In our end-to-end model, features learned from the CNN branch are combined with the extracted features from a pre-trained model with a concatenation layer. Loss values are then computed as the weighted sum of losses acquired from the branches. A combination of pre-trained features with the features learned from a traditional CNN yield more successful results than the classical single input models. Especially in the case of a small amount of data, feature extraction from a pre-trained model contributes to classification results. AlexNet was chosen as a feature extractor because of its reduced number of parameters compared with the performance of other pre-trained models. As a network with fewer parameters that reduces the training time and computational complexity, AlexNet is still one of the best models for feature extraction compared with the other models (Ahmed and Asif, 2020; Zhang et al., 2018).

The proposed model yields better results than the related studies. When the classification performance is compared with the related studies and are then analyzed vis-Ã -vis the results for the Plant Village dataset shown in Table 12, some of the presented studies seem to be more accurate compared to our model. The main reason for this is that the models in these studies were trained with data from a limited number of classes, and most of them were diseases within the same species. As shown in Table 12, our proposed method is superior to (Chen et al., 2020a; Li et al., 2020) which used exclusively maize diseases in their experiments. We trained our models with 17 different classes during the experiments. Moreover, some studies use a binary disease classification (Hernández and López, 2020) as healthy/non-healthy. For such cases, our model shows a higher and or comparable accuracy. For the FISB dataset, the proposed model is better than the compared studies (Chen et al., 2020a; Chen et al., 2020b). The images in the FISB dataset were captured in actual field settings, and therefore include a wide diversity of backgrounds and an extensive variety of symptom characteristics. The images were captured under non-uniform illumination intensities and clutter field background conditions. Most photographic materials in the commonly used datasets employed by similar studies include images captured solely in experimental (laboratory) setups such as Plant Village, rather than in real-life environments. The FISB dataset was used in our experiments to increase the generalizability of the proposed method, and it yielded strong results.

In future research, we plan to design new network models in cross-talk architecture for the same purpose. We are planning to design an end-to-end network model by embedding the pre-trained model into our new end-to-end model. The layers transferred from the pre-trained model will be frozen to keep their weight the same during the training process. We are also planning to extend our model with new plant types. We will propose a multi-modal CNN model that takes visual data of a plant leaf image and other non-visual parameters like climatic stress and environmental conditions.

Finally, to the best of our knowledge, there is as yet no multi-input multi-task model on plant disease prediction in the literature. Through this landmark study, we have confirmed through the ablation studies and experiments that multi-task learning is superior to single-task learning.

5. Conclusion

Plant disease detection and plant type classification problems are crucial and challenging problems in agriculture. We focused on the use of Multi-Task Learning and Transfer Learning approaches to solve these two related tasks at the same time. From the machine learning perspective, these two tasks are classification tasks, and we are therefore able to encode their representations in our proposed model. In this study, we proposed a novel multi-task supervised learning model for plant classification and disease detection tasks. We developed four models using two different datasets. While the first model classified data points into healthy and unhealthy classes, the second model specified the categories of the disease. The results of our experiments were remarkable, demonstrating the validity of our model. Additionally, our models used not only one type of image data; the pre-trained features were also incorporated into the multi-task neural network models. Although we used the Alexnet pre-trained model to obtain these additional features, our model is also conducive to other pre-trained models. We designed a CNN model from scratch using different types of layers, however, the presented CNN branch can be easily exchanged and adapted to different problems. Researchers in various fields can apply our proposed model to handle tasks in different application domains.

Declaration of Competing Interest

None.

References

Ahmed, N., Asif, H.M.S., 2020. Perceptual quality assessment of digital images using deep features. Computing and Informatics 39 (3), 385–409.

Albayrak, A., Çeven, S., Bayır, R., 2021. Modeling of migratory beekeeper behaviors with machine learning approach using meteorological and environmental variables: the case of Turkey. Ecological Informatics 66, 101470.

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M.G., Bereciartua, A., Alvarez-Gila, A., 2020. Few-shot learning approach for plant disease classification using images taken in the field. Comput. Electron. Agric. 175, 105542.

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., Stefanovic, D., 2019. Solving current limitations of deep learning based approaches for plant disease detection. Symmetry 11 (7), 939.

Bai, X., Cao, Z., Zhao, L., Zhang, J., Lv, C., Li, C., Xie, J., 2018. Rice heading stage automatic observation by multi-classifier cascade based rice spike detection method. Agric. For. Meteorol. 259, 260–270.

Baltrušaitis, T., Ahuja, C., Morency, L.-P., 2018. Multimodal machine learning: a survey and taxonomy. IEEE Trans. Pattern Anal. Mach. Intell. 41 (2), 423–443.

Brahimi, M., Boukhalfa, K., Moussaoui, A., 2017. Deep learning for tomato diseases: classification and symptoms visualization. Appl. Artif. Intell. 31 (4), 299–315.

Chen, J., Chen, J., Zhang, D., Sun, Y., Nanehkaran, Y.A., 2020a. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agric. 173, 105393.

Chen, J., Zhang, D., Nanehkaran, Y.A., Li, D., 2020b. Detection of rice plant diseases based on deep transfer learning. J. Sci. Food Agric. 100 (7), 3246–3256.

Ebrahimi, M., Khoshtaghaza, M.-H., Minaei, S., Jamshidi, B., 2017. Vision-based pest detection based on svm classification method. Comput. Electron. Agric. 137, 52–58.

Ferentinos, K.P., 2018. Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. 145, 311–318.

Golhani, K., Balasundram, S.K., Vadamalai, G., Pradhan, B., 2018. A review of neural networks in plant disease detection using hyperspectral data. Information Processing in Agriculture 5 (3), 354–371.

Govaerts, R., 2001. How many species of seed plants are there? Taxon 50 (4), 1085–1090.

Hernández, S., López, J.L., 2020. Uncertainty quantification for plant disease detection using bayesian deep learning. Appl. Soft Comput. 96, 106597.

Hughes, D., Salathé, M., et al., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060.

Kar, S., Purbey, V.K., Suradhaniwar, S., Korbu, L.B., Kholová, J., Durbha, S.S., Adinarayana, J., Vadez, V., 2021. An ensemble machine learning approach for determination of the optimum sampling time for evapotranspiration assessment from high-throughput phenotyping data. Comput. Electron. Agric. 182, 105992.

Kawasaki, Y., Uga, H., Kagiwada, S., Iyatomi, H., 2015. Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In: International Symposium on Visual Computing. Springer, pp. 638–645.

Kaya, A., Keceli, A.S., Catal, C., Yalic, H.Y., Temucin, H., Tekinerdogan, B., 2019. Analysis of transfer learning for deep neural network based plant classification models. Comput. Electron. Agric. 158, 20–29.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.

Lai, X., Wu, X., Zhang, L., 2021. Autoencoder-based multi-task learning for imputation and classification of incomplete data. Appl. Soft Comput. 98, 106838 https://doi.org/10.1016/j.asoc.2020.106838.

Li, Y., Nie, J., Chao, X., 2020. Do we really need deep cnn for plant diseases identification? Comput. Electron. Agric. 178, 105803.

Lu, Y., Yi, S., Zeng, N., Liu, Y., Zhang, Y., 2017. Identification of rice diseases using deep convolutional neural networks. Neurocomputing 267, 378–384.

Misra, I., Shrivastava, A., Gupta, A., Hebert, M., 2016. Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3994–4003.

Mora, C., Tittensor, D.P., Adl, S., Simpson, A.G., Worm, B., 2011. How many species are there on earth and in the ocean? PLoS Biol. 9 (8), e1001127.

Ruder, S., Bingel, J., Augenstein, I., Søgaard, A., 2019. Latent multi-task architecture learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 33, pp. 4822–4829.

Scotland, R.W., Wortley, A.H., 2003. How many species of seed plants are there? Taxon 52 (1), 101–104.

Sharma, P., Berwal, Y.P.S., Ghai, W., 2020. Performance analysis of deep learning cnn models for disease detection in plants using image segmentation. Information Processing in Agriculture 7 (4), 566–574.

Strezoski, G., van Noord, N., Worring, M., 2019. Learning task relatedness in multi-task learning for images in context. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval, pp. 78–86.

Tiwari, V., Joshi, R.C., Dutta, M.K., 2021. Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. Ecological Informatics 101289.

Vandenhende, S., Georgoulis, S., Van Gool, L., 2020. Mti-net: multi-scale task interaction networks for multi-task learning. In: European Conference on Computer Vision. Springer, pp. 527–543.

Wäldchen, J., Rzanny, M., Seeland, M., Mäder, P., 2018. Automated plant species identification—trends and future directions. PLoS Comput. Biol. 14 (4), e1005993.

Yang, W., Li, Z., Wang, C., Li, J., 2020. A multi-task faster r-cnn method for 3d vehicle detection based on a single image. Appl. Soft Comput. 95, 106533.

Yang, Z., He, W., Fan, X., Tjahjadi, T., 2022. Plantnet: transfer learning-based fine-grained network for high-throughput plants recognition. Soft. Comput. 1–10.

Yu, H., Cheng, X., Chen, C., Heidari, A.A., Liu, J., Cai, Z., Chen, H., 2022. Apple leaf disease recognition method with improved residual network. Multimed. Tools Appl. 1–24.

Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595.

Zhang, Y., Yang, Q., 2017. A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering.

Zhang, Y., Yang, Q., 2018. An overview of multi-task learning. Natl. Sci. Rev. 5 (1), 30–43.

Zhang, Z., Luo, P., Loy, C.C., Tang, X., 2014. Facial landmark detection by deep multi-task learning. In: European Conference on Computer Vision. Springer, pp. 94–108.

Zhao, X., Li, H., Shen, X., Liang, X., Wu, Y., 2018. A modulation module for multi-task learning with applications in image retrieval. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 401–416.

Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P., 2016. On large-batch training for deep learning: generalization gap and sharp minima arXiv preprint arXiv:1609.04836.