

Deep Feature Extraction for Detection of Tomato Plant Diseases and Pests based on Leaf Images

Yahya Altuntaş^{1*}, Adnan Fatih Kocamaz²

¹ Apricot Research Institute, Department of Agricultural Machinery and Technologies, 44090 Malatya, Turkey

² Inonu University, Department of Computer Engineering, 44280 Malatya, Turkey

* yahyaaltuntas@gmail.com

*Orcid: 0000-0002-7472-8251

Received: 19 October 2020

Accepted: 30 March 2021

DOI: 10.18466/cbayarfb.812375

Abstract

Plant diseases and pests cause yield and quality losses. It has great importance to detect plant diseases and pests quickly and with high accuracy in terms of preventing yield and quality losses. Plant disease and pest detection performed by plant protection experts through visual observation is a labor-intensive process with a high error rate. Developing effective, fast and highly successful computer-aided disease detection systems has become a necessity in terms of precision agriculture applications. In this study, well-known pre-trained convolutional neural network (CNN) models AlexNet, GoogLeNet and ResNet-50 are used as feature extractors. In addition, a deep learning model that concatenate deep features extracted from 3 CNN models has been proposed. The deep features were used to train the support vector machine classifier. The proposed model was used to classify leaf images of tomato plant diseases and pests, which is a subset of open-access PlantVillage dataset consisting of a total of 18835 images belonging to 10 classes including a healthy one. Accuracy, precision, sensitivity and f-score performance metrics were used with the hold-out validation method in determining model performances. Experimental results show that the detection of tomato plant diseases and pests is possible using concatenated deep features with an overall accuracy rate of 96.99%.

Keywords: convolutional neural networks, deep features extraction, deep learning, image classification, plant diseases and pests detection, precision agriculture.

1. Introduction

Tomato (*Solanum lycopersicum*) is one of the most widely cultivated agricultural products all over the world. As in all other agricultural products, one of the factors that negatively affect tomato cultivation is diseases and pests. Various diseases and pests cause yield and quality losses in crop production [1]. Early and accurate detection of diseases and pests is very important to prevent yield and quality losses. Plant disease and pest detection performed by visual observation by plant protection experts is a labor-intensive process with a high error rate [2]. The development of effective, fast and highly successful computer-aided disease detection systems has become a necessity for precision agriculture applications.

Remarkable progress has been made in plant disease and pest detection with studies based on traditional machine learning methods. Al-Hiary et al. [3] proposed a model for diagnosing 5 diseases that cause symptoms

in leaves in different plant species. They segmented the images using Otsu thresholding and k-means clustering methods and extracted texture features. They used these features to feed artificial neural networks (ANN) classifiers. Dubey and Jalal [4] performed feature extraction from apple images segmented by the k-means clustering method using local binary pattern methods to detect 3 different apple fruit diseases. They classified the obtained features using a support vector machine (SVM). Singh and Misra [5] conducted a study to detect 5 different diseases in 4 different plant species. They obtained the color co-occurrence matrix features from images that were segmented and enhanced with image processing techniques. According to their experimental results, they reported that the best performance was obtained with the SVM classifier. Success in traditional machine learning methods largely depends on the features used. Segmentation is required to extract the features. This situation necessitates that the work carried out with traditional machine learning methods should be under highly controlled conditions and limits the classification success achieved.

Deep learning refers to models that can learn the representations of data through multiple processing layers and thus perform end-to-end learning [6]. Thanks to deep learning, unlike traditional machine learning methods, learning from raw data can be performed without the need for feature extraction. In the literature, many studies have been conducted on the detection of plant diseases and pests based on deep learning [7, 8]. Mohanty et al. [1] retrained AlexNet and GoogLeNet pre-trained convolutional neural network (CNN) models, both from scratch and by fine-tuning. They reported that transfer learning provided faster converge on colored, gray-level and segmented images in their studies. Ferentinos [2] used deep learning methodology to identify plant diseases in his study on an open-access dataset containing 25 plant species and 58 plant-disease combinations. Although there are a limited number of studies on the detection of plant diseases and pests in more than one crop type, many studies have also been conducted on a single crop type such as apple [9], cucumber [10] and rice [11].

Fuentes et al. [12] proposed a deep learning-based approach to detect of tomato diseases and pests. They used a region-based CNN method to detect features on tomato leaf images. Durmuş et al. [13] retrained AlexNet and SqueezeNet pre-trained CNN models from scratch to detect tomato diseases and pests on the open-access PlantVillage dataset. Sardoğan et al. [14] used 500 healthy and infected tomato leaves from the PlantVillage dataset to detect 4 tomato diseases. They classified the features obtained from the fully connected layer of the proposed CNN architecture with the Learning Vector Quantization (LVQ) algorithm. Rangarajan et al. [15] fine-tuned both AlexNet and VGG16 pre-trained CNN models to detect of tomato plant diseases and pests. They analyzed both the role of the number of images and importance of hyperparameters in classification accuracy and execution time. Aversano et al. [16] fine-tuned the VGG19, Xception and ResNet-50 pre-trained CNN models for the detection of tomato diseases and pests.

Agarwal et al. [17] proposed a CNN architecture consisting of 3 convolution and 3 max pooling layers followed by 2 fully connected layers for the detection of tomato diseases and pest. Saeed et al. [18] used tomato, corn and potato leaves images from the PlantVillage dataset to build an automated crop disease recognition system. They selected the deep features extracted from the fully connected layers 6 and 7 of the VGG19 pre-trained CNN model using partial least squares (PLS) regression. They used selected deep features for model estimation using the ensemble baggag tree classifier.

In this study, the effect of using pre-trained CNN models as feature extractors on success in detecting tomato diseases and pests has been investigated. For this purpose, well-known pre-trained CNN models AlexNet, GoogLeNet and ResNet-50 were used as feature extractors. In addition, deep features fusion was performed by concatenating the deep features obtained from 3 CNN models. The obtained deep features were used to train the SVM classifier. According to the results, all CNN models achieved high classification success in detecting tomato diseases and pests by deep feature extraction. Concatenated deep features, on the other hand, achieved the best classification performance with an overall accuracy of 96.99%. The results of the experiments conducted within the scope of the study were compared with each other and with related studies in the literature.

The remaining of the paper is organized as follows: the materials and methods are given in Section 2. The experimental results are presented in Section 3. In Sections 4 and 5 discussion and conclusion remarks are given, respectively.

2. Materials and Methods

2.1. Dataset

In this study, experiments have been carried out on diseased and healthy tomato leaf images, which is a subset of the open-access PlantVillage [19] dataset.



Figure 1. Sample images in the dataset.

Table 1. Details of classes in the dataset

Class Name	Disease Scientific Name	Images (Number)		
		Training	Test	Total
Bacterial spot	Xanthomonas campestris pv. vesicatoria	1703	424	2127
Early blight	Alternaria solani	800	200	1000
Healthy	-	1273	318	1591
Late blight	Phytophthora infestans	1528	381	1909
Leaf mold	Fulvia fulva	800	200	1000
Septoria leaf spot	Septoria lycopersici	1417	354	1771
Spider mites	Tetranychus urticae	1342	334	1676
Target spot	Corynespora cassicola	1124	280	1404
ToMV	Tomato mosaic virus (ToMV)	800	200	1000
TYLCV	Begomovirus (Fam. Geminiviridae)	4287	1070	5357
Total		15074	3761	18835

TYLCV: Tomato yellow leaf curl virus

Since the link reported in the original article has been broken, the version republished by Geetharamani and Pandian [20] has been used. The dataset consists of 18835 images belonging to 10 classes. Images are 256-by-256 pixels resolution and colored. The image format is JPEG. The space occupied by the dataset in the disk is 321 MB.

Sample images from the dataset have been demonstrated in Figure 1. The sample images from left to right in the first row belong to bacterial spot, early blight, healthy, late blight, and leaf mold classes; the sample images in the second row belong to septoria leaf spot, spider mites, target spot, mosaic virus, and yellow leaf curl virus classes.

AlexNet, GoogLeNet and ResNet-50 CNN models take input images with a resolution of 227-by-227, 224-by-224 and 224-by-224 pixels respectively. For this reason, images have been resized to the specified resolutions.

The hold-out validation method has been used to make a one-to-one comparison of the models applied within the scope of the study. For this purpose, the dataset has been divided into training and test groups at a ratio of 4:1. All models have been trained with the same training images and tested with the same testing images. The classes of the dataset, the scientific names of the diseases, the class distribution of the samples, and the number of training and testing images are given in Table 1.

2.2. Convolutional neural networks

CNNs are deep learning models designed to automatically learn representations of data. A CNN architecture consists of two parts. The first part consists of convolution, activation and pooling layers where discriminative features of data is learned, the second

part consists of fully connected layers and softmax layer where the learned features are classified [6].

Thanks to the filters in the convolution layer, the interrelated spatial dependencies of the data are discovered. Filter weights are shared. In this way, it does not affect learning where the same discriminative feature in different locations of the input data [21]. As a result of the convolution process, the weighted sum of inputs is obtained. This layer is followed by an activation layer to get rid of linear dependencies. Although there are different activation functions such as hyperbolic tangent and sigmoid, the most preferred activation function is the Rectified Linear Unit (ReLU). ReLU activation function sets negative values to 0. Feature maps are obtained in this layer. Next comes the pooling layer to shrink in size, without losing valuable information in the data. A maximum or average pooling can be done. Convolution operations are repeated one after another depending on the hyper-parameter values such as input size, filter size, stride and padding. Ultimately, it is transferred to the fully connected layer. The purpose of this layer is to flatten the learned features. Depending on the architecture, there may be one or more fully connected layers. Then the last fully connected layer is passed to a softmax layer. Finally, the model estimation is made by performing class probability calculations of the features learned with the softmax layer. There are as many outputs as the number of classes in the problem addressed in the last fully connected layer.

2.3. Deep feature extraction and proposed model

Deep feature extraction is a transfer learning approach to employ a pre-trained CNN model for a similar task. Deep feature extraction is to use a pre-trained CNN model as a feature extractor. In this approach, the model parameters (weights) are used without fine-tuning [22].

Then a classification algorithm such as SVM is used for the new classification task [23]. In this approach, the pre-trained CNN model is employed as part of the solution of the new task using as a feature extractor.

In the deep feature extraction approach, deep features can be obtained from any layer of a pre-trained CNN model. However, common usage is to extract deep features from the last fully connected layer. In this study, the last fully connected layers of AlexNet, GoogLeNet and ResNet-50 CNN models are used to extract the deep features. In addition, a deep learning model that concatenate deep features obtained from 3 CNN models has been proposed to increase prediction performance. Overall structure of the proposed model is shown in Figure 2.

The pre-trained CNN models used in the study are briefly introduced below. AlexNet [24] is one of the pioneering CNN architecture. AlexNet has a simple architecture that consists of 5 convolution layers, and 3 fully connected layers.

GoogLeNet [25] is a 22 layer CNN architecture that's a variant of the inception network. The most important properties of GoogLeNet architecture is the use of the inception module. The module simply provides a direct link between multiple different layers. This increases network complexity while keeping the computing cost at the same level.

ResNet-50 [26] architecture consists of 50 layers. ResNet-50 is different from other architectures with its micro-architecture module structure. In architecture, it may be preferred to switch to the lower layer by ignoring the change between some layers. In ResNet architecture, the performance rate has been increased by allowing transition operations between blocks with this structure.

2.4. Support vector machines

The deep feature extraction approach requires a classifier method to be trained with the extracted deep features. In this study, SVM proposed by Vapnik [27] was used as a classifier. It has been reported that the SVM classifier shows superior performance in different agricultural image classification problems [28].

The basic principle in solving a classification problem with SVM is to determine a hyperplane that divides samples belonging to two classes optimally from each other. The formula for the output of a linear SVM is given in Equation (2.1), where \vec{w} is the normal vector to the hyperplane and \vec{x} is the input vector. Maximizing margins can be defined as an optimization problem: minimize Equation (2.2) subject to Equation (2.3) where y_i and \vec{x}_i are the correct output of the SVM and the input vector for i^{th} training sample, respectively [29].

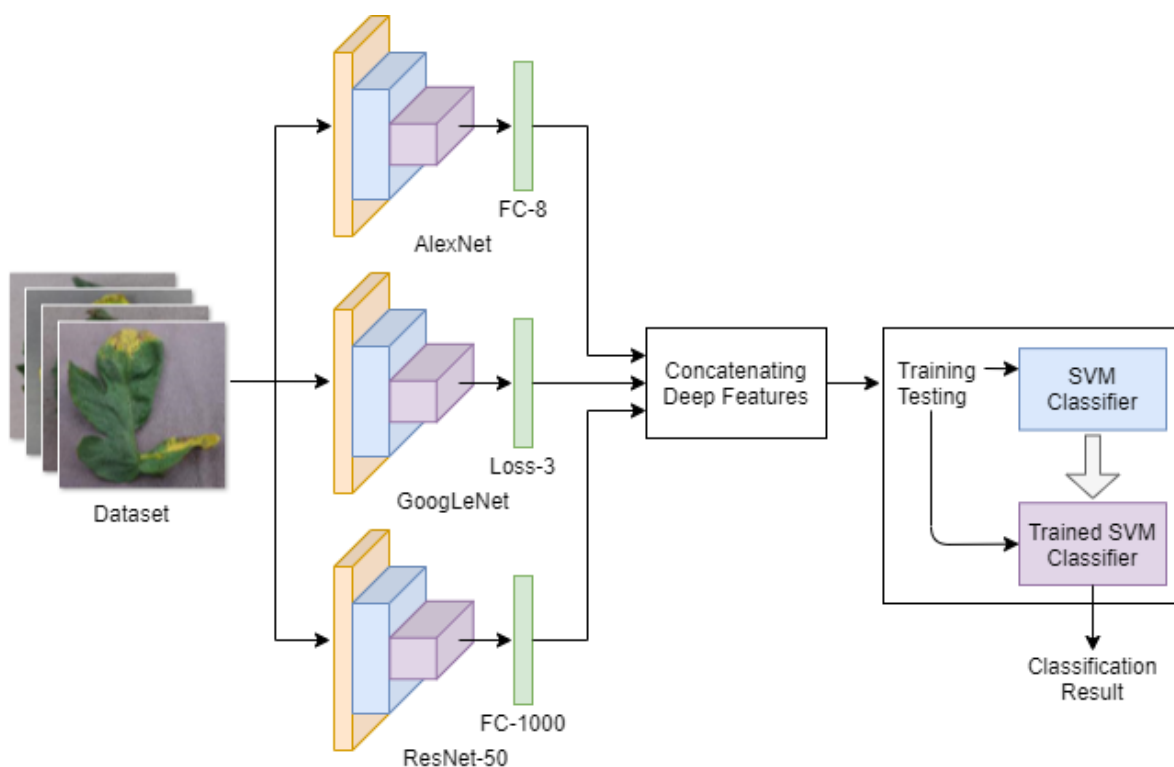


Figure 2. Overall structure of the proposed model

$$u = \vec{w} \cdot \vec{x} - b \quad (2.1)$$

$$\frac{1}{2} \|\vec{w}\|^2 \quad (2.2)$$

$$y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1, \forall i \quad (2.3)$$

$$Pre. = \frac{TN}{TN + FP} \quad (3.2)$$

$$Sen. = \frac{TP}{TP + FN} \quad (3.3)$$

$$F - score = \frac{2TP}{2TP + FP + FN} \quad (3.4)$$

SVM is a binary classifier defined to separate only 2 classes and do not support for multi-class classification problems. One strategy for multi-class classification with SVMs is to create a one-to-one set of classifiers and predict the class chosen by the majority of classifiers [30]. While this enables creating classifiers $K(K-1)/2$ for the classification problem with K classes, the training time of the classifiers may be reduced as the training data set for each classifier will be smaller.

3. Results

In this study, the deep feature extraction method, which is a transfer learning approach employing pre-trained CNN models as feature extractors, was used for the detection of tomato plant diseases and pests. The deep features were obtained from the last fully connected layers of AlexNet, GoogLeNet and ResNet-50 named FC-8, Loss-3 and FC-1000, respectively. In addition, a deep learning model that concatenate 1000 deep features obtained from each of the 3 CNN models proposed to increase predictive performance. Obtained deep features were used to train the SVM classifier. In the configuration of the SVM, all parameters are used in default configurations. Experimental studies were carried out on a computer with i5-8250U CPU, 8 GB RAM, 2 GB GPU, 256 SSD HDD hardware specifications. All experimental studies were implemented in MATLAB 2019b programming environment.

Since there is a sufficient number of images in the dataset, the hold-out validation method was used to make a one-to-one comparison of the models. For this purpose, the dataset was divided into training and test groups at a ratio of 4:1. Accuracy (Acc.), precision (Pre.), sensitivity (Sen.) and f-score performance metrics were calculated for comparison of model performances. These performance metrics are calculated with the indices true positive (TP), false negative (FN), false positive (FP), and true negative (TN) obtained from the confusion matrix. The mathematical equations of performance metrics used in comparing models are as follows:

$$Acc. = \frac{TP + TN}{TP + FN + FP + TN} \quad (3.1)$$

In multi-class classification problems, confusion matrix indices and performance metrics are calculated separately for each class. For the class for which performance metrics are calculated, samples belonging to that class are considered positive and all other samples are considered negative. Then, TP, FN, FP, and TN indices are calculated according to the predictions of the model for the samples treated as positive and negative.

While TP, FN, FP, and TN values are calculated for the Healthy class, samples labeled as Healthy in the dataset are considered positive and all other samples are considered negative. The number of positive samples in the test set, which are predicted as Healthy by the model, gives the TP value. The number of positive samples in the test set that are not predicted as Healthy by the model gives the FN value. The number of negative samples in the test set that are predicted as Healthy by the model gives the FP value. The number of negative samples in the test set that are not predicted as Healthy by the model gives the TN value.

That is, all samples except the class whose confusion matrix indices are calculated are considered negative and are named TN regardless of whether they are correctly predicted in their own class. This situation causes high accuracy and sensitivity values.

In multi-class classification problems, the overall accuracy metric is also used to measure the overall performance of the model. This metric gives us the ratio of the number of correct predictions of the model to the total number of predictions. It is calculated as follows.

$$Overall Acc. = \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FP_i)} \quad (3.5)$$

In the formula, N refers to the number of classes. TP_i and FP_i show the correct and incorrect prediction numbers of the i^{th} class respectively.

The classification results are given in Table 2. The values given in the table are confusion matrix (TP, FN, FP, and TN), Acc., Pre., Sen., F-Score (F-scr), and overall accuracy (Overall Acc.).

Table 2. Classification results

Pre-trained CNN Model	Class	TP	FN	FP	TN	Acc. (%)	Pre.	Sen.	F-Scr	Overall Acc. (%)
AlexNet	Bacterial spot	403	18	21	3319	98.96	0.957	0.993	0.953	92.18
	Early blight	153	50	47	3511	97.42	0.753	0.986	0.759	
	Healthy	304	11	14	3432	99.33	0.965	0.995	0.960	
	Late blight	348	38	33	3342	98.11	0.901	0.990	0.907	
	Leaf mold	168	23	32	3538	98.53	0.879	0.991	0.859	
	Sep. leaf spot	324	39	30	3368	98.16	0.892	0.991	0.903	
	Spider mites	295	39	39	3388	97.92	0.883	0.988	0.883	
	Target spot	242	55	38	3426	97.52	0.814	0.989	0.838	
	ToMV	186	6	14	3555	99.46	0.968	0.996	0.948	
	TYLCV	1044	15	26	2676	98.90	0.985	0.990	0.980	
GoogLeNet	Bacterial spot	390	37	34	3300	98.11	0.913	0.989	0.916	89.31
	Early blight	142	67	58	3494	96.67	0.679	0.983	0.694	
	Healthy	300	16	18	3427	99.09	0.949	0.994	0.946	
	Late blight	329	41	52	3339	97.52	0.889	0.984	0.876	
	Leaf mold	168	29	32	3532	98.37	0.852	0.991	0.846	
	Sep. leaf spot	294	66	60	3341	96.64	0.816	0.982	0.823	
	Spider mites	289	49	45	3378	97.50	0.855	0.986	0.860	
	Target spot	227	57	53	3424	97.07	0.799	0.984	0.804	
	ToMV	181	12	19	3549	99.17	0.937	0.994	0.921	
	TYLCV	1039	28	31	2663	98.43	0.973	0.988	0.972	
ResNet-50	Bacterial spot	414	7	10	3330	99.54	0.983	0.997	0.979	96.96
	Early blight	173	20	27	3541	98.75	0.896	0.992	0.880	
	Healthy	313	3	5	3440	99.78	0.990	0.998	0.987	
	Late blight	368	21	13	3359	99.09	0.946	0.996	0.995	
	Leaf mold	193	6	7	3555	99.65	0.969	0.998	0.967	
	Sep. leaf spot	344	10	10	3397	99.46	0.971	0.997	0.971	
	Spider mites	324	21	10	3406	99.17	0.939	0.997	0.954	
	Target spot	259	18	21	3463	98.96	0.935	0.993	0.929	
	ToMV	194	3	6	3558	99.76	0.984	0.998	0.977	
	TYLCV	1065	5	5	2686	99.73	0.995	0.998	0.995	
Concatenated Deep Features	Bacterial spot	419	4	5	3333	99.76	0.990	0.998	0.989	96.99
	Early blight	175	22	25	3539	98.75	0.888	0.992	0.881	
	Healthy	312	5	6	3438	99.70	0.984	0.998	0.982	
	Late blight	366	15	15	3365	99.20	0.960	0.995	0.960	
	Leaf mold	188	8	12	3553	99.46	0.959	0.996	0.949	
	Sep. leaf spot	345	16	9	3391	99.33	0.955	0.997	0.965	
	Spider mites	326	14	8	3413	99.41	0.958	0.997	0.967	
	Target spot	260	18	20	3463	98.98	0.935	0.994	0.931	
	ToMV	196	6	4	3555	99.73	0.970	0.998	0.975	
	TYLCV	1061	5	9	2686	99.62	0.995	0.996	0.993	

According to the experimental results, the overall accuracy rates obtained by classifying the deep features extracted from the last fully connected layers of AlexNet, GoogLeNet and ResNet-50 pre-trained CNN models with SVM classifier are 92.18%, 89.31% and 96.96%, respectively. According to the experimental results, all CNN models used within the scope of the study showed superior classification performance. However, the best performance has been achieved by using concatenating deep features with an overall accuracy rate of 96.99%. This overall accuracy rate is better than all CNN models individually.

The time taken to extract deep features for 3761 test images and store the resulting deep feature vectors on disk as a separate file was measured as 14 minutes 58 seconds, 20 minutes 19 seconds, and 50 minutes 33 seconds for AlexNet, GoogLeNet and ResNet-50 models, respectively. Experimental results show that deep feature extraction process from CNN models with a high number of layers requires more time.

Table 3. Comparison of the proposed model and the related studies

Related studies	Crops type	Method	Overall Acc. (%)
Fuentes et al., 2017 [12]	Tomato (10 classes)	Region-based CNN	83.60
Durmuş et al., 2017 [13]	Tomato (10 classes)	Retraining a pre-trained CNN model from scratch	95.65
Sardoğan et al., 2018 [14]	Tomato (5 classes)	CNN with LVQ algorithm	86.00
Rangarajan et al., 2018 [15]	Tomato (7 classes)	Fine-tuning a pre-trained CNN model	97.49
Aversano et al., 2020 [16]	Tomato (10 classes)	Fine-tuning a pre-trained CNN model	97.16
Agarwal et al., 2020 [17]	Tomato (10 classes)	CNN	91.20
Saeed et al., 2021 [18]	Tomato (10 classes)	PLS-based deep features selection and ensemble baggage tree classifier	87.11
	Corn (4 classes)		91.67
	Potato (3 classes)		91.67
This study	Tomato (10 classes)	Concatenating deep features and SVM	96.99

4. Discussion

The number of classes dealt with in the related studies is not equal. Sardoğan et al. [14] conducted a study with a total of 5 classes, including 4 diseases and 1 healthy class. Rangarajan et al. [15] conducted a study with a total of 7 classes, including 6 diseases and 1 healthy class. In other related studies, 10 classes, including 9 diseases and 1 healthy class were conducted. Since test samples are not pre-defined in the PlantVillage dataset, the test samples used in related studies are not the same. Also, the number of test samples used is not equal. Sardoğan et al. [14] in their work with a total of 500 images, they used 400 images for training and 100 images for testing. Agarwal et al. [17] in their work with a total of 17500 images, they used 10000 images for training, 7000 images for validation and 500 images for testing. A one-to-one comparison among the related studies is not feasible due to the reasons explained above. Nevertheless, we present a comparison in Table 3 considering several criteria, such as the number of classes, methods, and overall accuracy rates.

5. Conclusion

In this study, the possibilities of detecting tomato plant diseases and pests by deep feature extraction were investigated. For this purpose, well-known pre-trained CNN models AlexNet, GoogLeNet, and ResNet-50 were used as feature extractors. 1000 deep features obtained from the last fully connected layers of CNN models were used to train the SVM classifier. In addition, a deep learning model that concatenate 1000 deep features obtained from each of the 3 CNN models proposed to increase predictive performance. According to the experimental results, while superior classification performance was obtained with deep features extracted from all CNN models, the best result was obtained with the concatenated deep features. The reason for this is considered to be the result of CNN models with different architectures discovering different discriminative features.

In future studies, it is aimed to increase the classification performance with different fusion methods. In addition, the performance of the models to be developed on field condition images will be evaluated.

Author's Contributions

Yahya Altuntaş: Drafted and wrote the manuscript, performed the experiment and result analysis.

Adnan Fatih Kocamaz: Supervised the experiments' progress, result interpretation and helped in manuscript preparation.

Ethics

There are no ethical issues after the publication of this manuscript.

References

1. Mohanty, SP, Hughes, DP, Salathé, M. 2016. Using deep learning for image-based plant disease detection. *Frontiers in plant science*; 7: 1419.
2. Ferentinos, KP. 2018. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*; 145: 311-318.
3. Al-Hiary, H, Bani-Ahmad, S, Reyalat, M, Braik, M, Alrahamneh, Z. 2011. Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications*; 17(1): 31-38.
4. Dubey, SR, Jalal, AS. Detection and classification of apple fruit diseases using complete local binary patterns, proceedings of the 3rd international conference on computer and communication technology, Allahabad, India, 2012, pp 346-351.
5. Singh, V, Misra, AK. 2017. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information processing in Agriculture*; 4(1): 41-49.
6. LeCun, Y, Bengio, Y, Hinton, G. 2015. Deep learning. *Nature*; 521(7553): 436-444.
7. Ünal, Z. 2020. Smart Farming Becomes Even Smarter With Deep Learning - A Bibliographical Analysis. *IEEE Access*; 8:105587-105609.

8. Kamilaris, A, Prenafeta-Boldú, FX. 2018. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*; 147: 70-90.
9. Zhong, Y, Zhao, M. 2020. Research on deep learning in apple leaf disease recognition. *Computers and Electronics in Agriculture*; 168: 105146.
10. Fujita, E, Kawasaki, Y, Uga, H, Kagiwada, S, Iyatomi, H. Basic investigation on a robust and practical plant diagnostic system, proceedings of the 15th IEEE International Conference on Machine Learning and Applications, Anaheim CA, USA, 2016, pp 989-992.
11. Lu, Y, Yi, S, Zeng, N, Liu, Y, Zhang, Y. 2017. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*; 267: 378-384.
12. Fuentes, A, Yoon, S, Kim, SC, Park, DS. 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*; 17(9): 2022.
13. Durmuş, H, Güneş, EO, Kırıcı, M. Disease detection on the leaves of the tomato plants by using deep learning, proceedings of the 6th International Conference on Agro-Geoinformatics, Fairfax VA, USA, 2017, pp 1-5.
14. Sardogan, M, Tuncer, A, Ozen, Y. Plant leaf disease detection and classification based on CNN with LVQ algorithm, proceedings of the 3rd International Conference on Computer Science and Engineering, Sarajevo, Bosnia and Herzegovina, 2018, pp 382-385.
15. Rangarajan, AK, Purushothaman, R, Ramesh, A. 2018. Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia computer science*; 133: 1040-1047.
16. Aversano, L, Bernardi, ML, Cimitile, M, Iammarino, M, Rondinella, S. Tomato diseases Classification Based on VGG and Transfer Learning, proceedings of the IEEE International Workshop on Metrology for Agriculture and Forestry, Virtual Conference, 2020, pp 129-133.
17. Agarwal, M, Singh, A, Arjaria, S, Sinha, A, Gupta, S. 2020. ToLeD: Tomato leaf disease detection using convolution neural network. *Procedia Computer Science*; 167: 293-301.
18. Saeed, F, Khan, MA, Sharif, M, Mittal, M, Goyal, LM, Roy, S. 2021. Deep neural network features fusion and selection based on PLS regression with an application for crops diseases classification. *Applied Soft Computing*; 103: 107164.
19. Hughes, D, Salathé, M. 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint*; arXiv:1511.08060.
20. Geetharamani, G, Pandian, A. 2019. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Computers & Electrical Engineering*; 76: 323-338.
21. Altuntaş, Y, Cömert, Z, Kocamaz, AF. 2019. Identification of haploid and diploid maize seeds using convolutional neural networks and a transfer learning approach. *Computers and Electronics in Agriculture*; 163: 104874.
22. Chatfield, K, Simonyan, K, Vedaldi, A, Zisserman, A. 2014. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint*; arXiv:1405.3531.
23. Toğaçar, M, Ergen, B, Cömert, Z. 2020. Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models. *Measurement*; 158: 107703.
24. Krizhevsky, A, Sutskever, I, Hinton, GE. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*; 25: 1097-1105.
25. Szegedy, C, Liu, W, Jia, Y, Sermanet, P, Reed, S, Anguelov, D, Rabinovich, A. Going deeper with convolutions, proceedings of the IEEE conference on computer vision and pattern recognition, Boston MA, USA, 2015, pp 1-9.
26. He, K, Zhang, X, Ren, S, Sun, J. Deep residual learning for image recognition, proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas NV, USA, 2016, pp 770-778.
27. Vapnik, VN. The nature of statistical learning theory; Springer, New York, USA, 2000; pp 313.
28. Altuntaş, Y, Kocamaz, AF. 2019. Renk momentleri ve destek vektör makineleri kullanarak haploid mısır tohumlarının tanımlanmasında renk uzaylarının sınıflandırma performansına etkisinin karşılaştırılması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*; 31(2): 551-560.
29. Huang, S, Cai, N, Pacheco, PP, Narrandes, S, Wang, Y, Xu, W. 2018. Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics-Proteomics*; 15(1): 41-51.
30. Bishop, CM. Pattern recognition and machine learning; Springer, New York, USA, 2006; pp 738.