Unsupervised Convolutional Autoencoder-Based Feature Learning for Automatic Detection of Plant Diseases

Hilman F. Pardede, Endang Suryawati, Rika Sustika, and Vicky Zilvan
Research Center for Informatics
Indonesian Institute of Sciences, Bandung, Indonesia
Email:{hilm001, enda029, rika002, vick001@}@lipi.go.id

***Abstract*—Developing an automatic detector of plant diseases is one of application fields in machine learning. Ground-truth diagnoses of plant diseases which are conducted by experts in laboratory tests are often inapplicable for fast and cheap implementations. Using machine learning approaches, the images of leaves or fruits are used as input data. From the data, we design discriminative features that are good for diseases classification. However, finding suitable features from the images are often challenging due to high intra-variability and inter-variability of the data. In this paper, we present an unsupervised feature learning algorithm using the convolutional autoencoder for detection of plant diseases. The use of convolutional autoencoder has two main advantages. First, the use of handcrafted features is not necessary as the network itself may learn to produce discriminative features. Secondly, the procedure is conducted in an unsupervised manner and hence, no labeling of the data are required. Here, we use the output of the autoencoder as inputs to SVM-based classifiers for automatic detection of plant diseases. The method indicates to be better than conventional autoencoder with more hidden layers.**

***Index Terms*—Plant diseases detection, convolutional autoencoder, feature learning, SVM**

I. Introduction

Automatic plant diseases detection is one implementation of artificial intelligence-based machine learning [1], [2]. This application is needed due to the expensive cost of diagnosing of plant diseases using laboratory tests which is not affordable by smallholder farmers. Using machine learning, the images of the leaves and/or fruits are commonly used and then the features are extracted from the image data. Developing machine learning-based detector is quite challenging. One of the challenges is high intra and inter variability of the data. The diseases of the plants could be detected by the existence the spots or rotten areas on the leaves. Spots or the rotten areas may vary for the same diseases or some diseases may appear to have similar symptoms [3].

The process of finding the correct prediction in machine learning could be decomposed intro three sub-processes. First is the extraction of features. The purpose is to extract valuable information and discard unwanted one. Second is feature selection/reduction. In many applications, the features have high dimensions, and hence it may be better to select most appropriate features. Third is pattern finding where given a particular machine learning method, we aim to find the most suitable hyperparameters of the method that best fit the data with the class labels.

In conventional machine learning, feature extraction process should be able to produce features that are able to discriminate the classes well. In other words, they are separable in feature space. Handcrafted features, where the process are designed such that the features are discriminative in feature space. Therefore many plant diseases detection systems requires complex feature extraction processes [1], [4], [5], [6], or expensive cameras [7].

Deep learning is recently introduced in machine learning. Deep learning methods are based on artificial neural network (ANN) [8]. The difference is the use of various non-linear feature transformation unlike conventional neural networks where single nonlinear function is used. As the consequences, deep learning is able to learn complex nonlinear relations between the features and class labels. Deep learning is currently the dominant technology of object recognition [9]. Plant diseases detection could be seen as an examples of object recognition tasks. Some studies have proposed to use deep learning for plant diseases detection [10], [11]. In these studies, deep learning is used for supervised learning.

Currently, many studies try to use deep learning for unsupervised learning. The networks are designed to map the data to itself with the purpose for the networks to learn the better features by themselves. This is called feature or representation learning [12], [13]. One advantage of this approach is there is no need to manually label the data and as the consequences, we can train the networks with great amount of unlabeled data and use only small amount of data to train classifiers with supervised learning. This approach produces satisfactory results in many studies [14], [15]. To the best of our knowledge, no effort has been made to investigate unsupervised learning for plant diseases detection.

Autoencoder is an examples of unsupervised deep learning methods [16]. However, standard autoencoder may not perform well when data variability is high or data are corrupted by noises. Denoising autoencoder [17] and variantional autoencoder [18] are varians of autoencoder to tackle these issues. In addition, autoencoder, which is usually build from stacked fully connected networks may have limitations on

978-1-5386-5741-6/18/$31.00 ©2018 IEEE

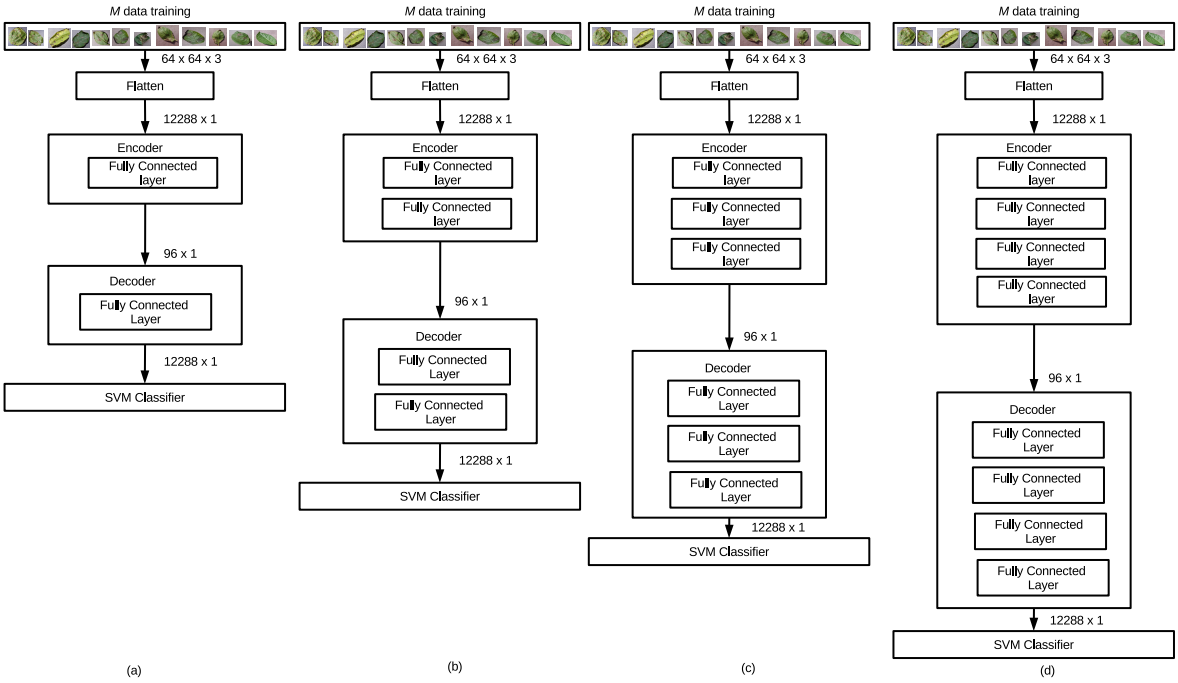

Fig. 1. The architectures of autoencoders built with stacked fully connected networks that are used in this paper. We denote it (a) Auto1, (b) Auto2, (c) Auto3, and (d) Auto4

image data. Autoencoder with convolutional layers [19] which have more emphasis on local relation between inputs more than a fully connected network may be more suitable for image data as in our tasks. In this paper we propose unsupervised feature learning-based convolutional autoencoder for plant diseases detection tasks. The output of our convolutional autoencoder is then used for SVM classifier.

The paper is organised as follows. In Sec. II, we briefly describe autoencoders. The the proposed method is explained in Sec. III. The experimental configurations are shown in Sec. IV and the results are presented and discussed in Sec. V. The paper is concluded in Sec. VI

II. Autoencoders

An autoencoder is a network consist of an encoder that maps data from high dimensional space into low-dimensional space and a decoder that transforms the data back into high dimensional space [26]. This can be done by building a network where the number of the output nodes are much smaller than the input nodes. Autoencoders are trained in unsupervised manner and can be seen as a non-linear feature extraction method. When autoencoders with large number of hidden layers are used (deep autoencoder), the training using back-propagation becomes ineffective. Usually a deep autoencoder trained by training each layer as an autoencoder with single layer [20]. This is called pre-training.

We develop several architectures of autoencoder for our tasks (See Fig 1). We develop 4 autoencoder architectures. For notation purposes, we call them Auto1, Auto2, Auto3, and Auto4. In first architecture (Auto1), we design a network where the encoder maps the input into 96 output nodes and the decoder transform them back into 12288 nodes. Then, we add more hidden layers to see the effect of adding more layers to the performance of the systems. For Auto2, the encoder are designed to have 1536 output nodes on the first hidden layer and 96 nodes for the second hidden layer while the decoder map them back to 1536 and 12288 nodes. For Auto3, the hidden layers of the encoder have 1536, 768, and 96 output nodes and the hidden layers of the decoder have 768, 1536, and 12288 output nodes. Lastly, for Auto4, we use networks with 4 hidden layers for the encoder and 4 hidden layers for the decoder with structures of mapping to 3072, 1536, 768, and 98 respectively.

III. The Proposed Method

In many applications of deep learning for image data, a fully connected networks, where the nodes of the inputs are connected to all nodes of the output, may not be optimal [21]. Usually, fully connected layers are replaced by convolutional layers for image data. A typical convolution layer consist of a convolution operation and pooling. For convolution operation, the nodes of the outputs are only connected to a particular inputs. In a sense, a convolutional layers try to find a nonlinear relations between neighborhood inputs first and the global relations are learned when data pass through the higher networks. This allow the network to learn “different latent representations” while the data pass to different layers of the networks. A pooling layer is usually included after the convolutional operation. Pooling layer works by combining

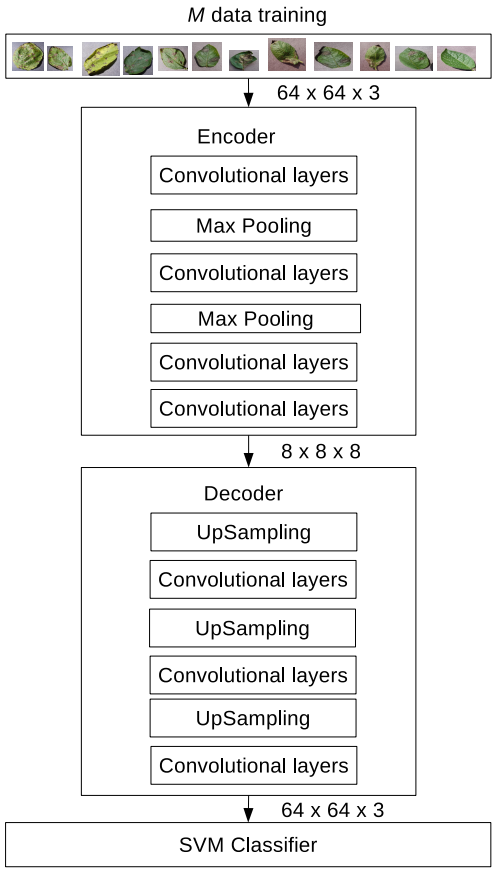

Fig. 2. The architecture of our convolutional autoencoder

the grouped neuron to produce a single output. In other words, pooling layer reduces the dimensionality of the input data.

The architecture of the convolutional autoencoder is shown in Fig. 2. For encoder, we use 4 layers of convolutional layers where max-pooling layers are applied in the first and second layers. For decoders, Upsampling layers and convolutional layers are applied to transform the data back to the original dimensions. We denote this architecture as Proposed1. We also apply regularization to avoid overfitting. For each convolutional layer, we apply $\ell 2$ regularization (with the value is set to 5×10^{10}). We denote this as Proposed2.

IV. Experimental Setup

Subset of corn and potato plants from Plantvillage dataset is used in this experiments [22]. A total of 6,004 data are selected. The distribution of data for each class is shown in Table I. The size of the images is reduced into 64×64 from the original size 256×256. We split the data into 80 % data for training and 20 % data testing. We train all autoencoders using unsupervised learning using the same data that will be used for training the SVM classifiers. We notice that the systems would be improved when we use all data for training the encoders. But for fairness, we only report the results using the same data data for training the autoencoders and the SVM. We train the system independently for each plant. The reported accuracies in the paper are the average accuracies over all classes (four classes for corn and three classes for potato).

For training, we vary number of epoch: 100, 150, and 200. The autoencoders are used as feature extractor for SVM classifier in this study. We use linear, RBF, and polynomial (order 2 and 3) kernels for SVM and penalty parameters are set to 1 for all kernels. Accuracy is used to evaluate the performance of the classifiers.

TABLE I
Data distributions for the experiments

Plants	Diseases	Number of data
Corn	Corn Gray Leaf Spot	513
	Corn Common Rust	1192
	Corn Nothern Leaf Blight	985
	Healthy	1162
Potato	Potato Early Blight	1000
	Potato Late Blight	1000
	Healthy	152

TABLE II
The performance of Proposed1 when the number of epochs are varied.

Plants	SVM kernels	Number of epochs		
		100	150	200
Potato	Linear	82.83	**87.01**	84.92
	RBF	82.60	**83.29**	82.83
	Polynomial (order 2)	80.05	81.67	**81.90**
	Polynomial (order 3)	74.48	75.87	**77.96**
Corn	Linear	**81.84**	80.42	79.77
	RBF	75.75	75.88	**76.01**
	Polynomial (order 2)	74.97	75.36	**75.75**
	Polynomial (order 3)	72.63	72.37	**73.02**

TABLE III
The performance comparison of Proposed1 and Proposed2. Experiments are conducted using 150 number of epochs to train the autoencoder.

Plants	SVM kernels	Proposed1	Proposed2
Potato	Linear	**87.01**	86.54
	RBF	83.29	**83.99**
	Polynomial (order 2)	81.67	**83.06**
	Polynomial (order 3)	75.87	**79.58**
Corn	Linear	**80.42**	78.60
	RBF	75.88	**76.13**
	Polynomial (order 2)	75.36	**75.88**
	Polynomial (order 3)	72.37	**73.02**

TABLE IV
THE PERFORMANCE COMPARISON OF PROPOSED METHODS WITH AUTOENCODERS BUILT WITH FULLY CONNECTED NETWORKS (I.E. AUTO1, AUTO2, AUTO3, AND AUTO4). EXPERIMENTS ARE CONDUCTED USING 150 NUMBER OF EPOCHS TO TRAIN THE AUTOENCODER.

Plants	SVM kernels	Auto1	Auto2	Auto3	Auto4	Proposed1	Proposed2
Potato	Linear	79.12	79.35	75.41	76.80	**87.01**	86.54
	RBF	73.55	75.41	74.71	76.57	83.29	**83.99**
	Polynomial (order 2)	73.32	74.94	74.48	76.80	81.67	**83.06**
	Polynomial (order 3)	74.01	74.48	75.17	47.10	75.87	**79.58**
Corn	Linear	77.30	81.84	75.88	75.49	**80.42**	78.60
	RBF	**76.65**	75.49	75.62	75.10	75.88	76.13
	Polynomial (order 2)	74.71	74.71	74.06	73.80	75.36	**75.88**
	Polynomial (order 3)	72.50	71.60	70.82	70.69	72.37	**73.02**

V. RESULTS AND DISCUSSIONS

A. Increasing Number of Epochs

Table II shows the performance of Proposed1 when the number of epochs are varied. In most cases, linear kernel achieves the best accuracy when we varied the number of epochs. Over all systems that we evaluate, the best performance is achieved when we set the number of epoch to 150 using linear kernel. Increasing the number of epoch does not necessarily improves the performance except when we use Polynomial kernels. However, the improvement of using Polynomial kernels is only slightly. The results are not surprising since adding more epochs could cause the system to overfit, degrading the performance of the systems. This is especially the case when nonlinear classifiers such as RBF kernels are used. Need to be noted that there is another parameter that needs to be optimized for RBF kernel, i.e. the gamma parameter. This would be in our future studies.

B. Adding Regularisation

The we compare Proposed1 and Proposed2 (the conditions where we add an $\ell 2$ regularization in training the convolutional layers). For these experiments, the number of epochs is set to 150. The results are presented in Table III. In most cases, adding regularization improves the performance except when linear kernels are used. This is not surprising since adding regularization would reduce the effect of over-fitting. As we can see, the benefit of regularization can be observed when we use nonlinear kernels such as RBF and polynomial kernels.

C. Comparison with Fully Connected Autoencoder

Table IV compares the proposed method with conventional autoencoders, i.e. the autoencoders built using stacked of fully connected networks. We compare the methods with autoencoders with various number of hidden layers (from one to four hidden layers). Our methods (Proposed1 and Proposed2) are generally better than conventional autoencoder especially when we use RBF and linear kernels where the methods achieve significant improvements compared to Auto1, Auto2, Auto3, and Auto4. Need to be noted, our convolutional autoencoders have the same hidden layers as Auto4.

We notice that for autoencoders with fully connected networks, adding more hidden layers does not improve the performance of the systems and in some cases, adding more layers may cause the performance to be worse. We observe, having only two hidden layers may be the optimum for this task.

VI. CONCLUSIONS

We propose convolutional autoencoders for unsupervised feature learning for plant diseases detection tasks. The output of the autoencoders are then used as input for SVM classifiers. Our experiments confirm that using convolutional layers are more effective than using a fully connected networks for this task. In the future, we would like to evaluate the robustness of using unsupervised feature learning against noisy and unseen data. We also would like to investigate various deep learning architectures for improving the performance of our deep feature learning architectures.

ACKNOWLEDGMENT

The paper is partially funded by Insinas Grant 2018 (Contract Number: 44/P/RPL-LIPI/INSINAS-1/III/2018) from Indonesian Ministry of Research, Technology, and Higher Education. The experiments on this research are conducted on High Performance Computing (HPC) facility in Research Center for Informatics, Indonesian Institute of Sciences.

REFERENCES

[1] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, "A review of advanced techniques for detecting plant diseases," *Computers and Electronics in Agriculture*, vol. 72, no. 1, pp. 1 – 13, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169910000438

[2] F. Martinelli, R. Scalenghe, S. Davino, S. Panno, G. Scuderi, P. Ruisi, P. Villa, D. Stroppiana, M. Boschetti, L. R. Goulart, C. E. Davis, and A. M. Dandekar, "Advanced methods of plant disease detection. a review," *Agronomy for Sustainable Development*, vol. 35, no. 1, pp. 1–25, Jan 2015. [Online]. Available: https://doi.org/10.1007/s13593-014-0246-1

[3] R. Kaundal, A. S. Kapoor, and G. P. Raghava, "Machine learning techniques in disease forecasting: a case study on rice blast prediction," *BMC bioinformatics*, vol. 7, no. 1, p. 485, 2006.

[4] D. J. MacKenzie, M. A. McLean, S. Mukerji, and M. Green, "Improved rna extraction from woody plants for the detection of viral pathogens by reverse transcription-polymerase chain reaction," *Plant disease*, vol. 81, no. 2, pp. 222–226, 1997.
[5] A. Y. Khaled, S. Abd Aziz, S. K. Bejo, N. M. Nawi, I. A. Seman, and D. I. Onwude, "Early detection of diseases in plant tissue using spectroscopy–applications and limitations," *Applied Spectroscopy Reviews*, vol. 53, no. 1, pp. 36–64, 2018.
[6] M. R. Badnakhe and P. R. Deshmukh, "An application of k-means clustering and artificial intelligence in pattern recognition for crop diseases," in *International Conference on Advancements in Information Technology*, 2011.
[7] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance," *Computers and Electronics in Agriculture*, vol. 74, no. 1, pp. 91–99, 2010.
[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.
[9] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.
[10] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
[11] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in plant science*, vol. 7, p. 1419, 2016.
[12] M. aurelio Ranzato, Y. lan Boureau, and Y. L. Cun, "Sparse feature learning for deep belief networks," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 1185–1192. [Online]. Available: http://papers.nips.cc/paper/3363-sparse-feature-learning-for-deep-belief-networks.pdf
[13] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
[14] S.-Z. Su, Z.-H. Liu, S.-P. Xu, S.-Z. Li, and R. Ji, "Sparse auto-encoder based feature learning for human body detection in depth image," *Signal Processing*, vol. 112, pp. 43–52, 2015.
[15] T. A. Lasko, J. C. Denny, and M. A. Levy, "Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data," *PloS one*, vol. 8, no. 6, p. e66341, 2013.
[16] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8595–8598.
[17] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1759–1763.
[18] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *European Conference on Computer Vision*. Springer, 2016, pp. 835–851.
[19] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep features learning for medical image analysis with convolutional autoencoder neural network," *IEEE Transactions on Big Data*, no. 1, pp. 1–1, 2017.
[20] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in neural information processing systems*, 2007, pp. 153–160.
[21] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.
[22] D. Hughes, M. Salathé *et al.*, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.