International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Crop Disease Detection Using Deep Learning Model

Mrs. Jahina J[1], Sathishkumar S[2], Poornavalli P R[3], Rithik P[4]

[1]ME. Assistant Professor, Department of AI and DS, Nandha Engineering College, Erode, India
[2,3,4]Department of AI and DS, Nandha Engineering College, Erode, India

Abstract

Detecting diseases in crops is a vital yet labor-intensive task in agriculture, often demanding extensive time and expert knowledge. This paper presents an innovative approach to crop disease detection using advanced computer vision and machine learning techniques. By automating the identification of common crop diseases, this system aims to reduce the reliance on expert intervention, expedite the diagnosis process, and ultimately improve crop management efficiency. The proposed method integrates deep learning models trained on a diverse dataset of diseased and healthy crop images, achieving high accuracy in disease recognition. This approach not only saves time but also provides farmers with a powerful tool to protect their crops from potential threats, thereby contributing to increased agricultural productivity and sustainability.

Keywords: Crop Disease Detection, Computer Vision, Machine Learning, Deep Learning, Agriculture, Automated Diagnosis, Precision Agriculture, Crop Management, Image Classification, Smart Farming.

INTRODUCTION

Recent advancements in computer vision and machine learning have revolutionized the field of agriculture, offering new tools to address the age-old challenge of crop disease detection. Traditionally, identifying diseases in crops has been a time-consuming and labor-intensive process, requiring the expertise of trained agronomists and pathologists. However, with the advent of smart technologies, it is now possible to automate this process, significantly reducing the time and effort required while improving accuracy.

In the context of agriculture, timely and accurate disease detection is crucial for preventing the spread of infections and minimizing crop losses. The application of machine learning models, particularly deep learning, enables the analysis of vast amounts of crop images to identify patterns and symptoms of diseases that may not be easily discernible to the human eye. By leveraging these technologies, farmers can receive real-time insights into the health of their crops, allowing for more precise interventions and better resource management.

The project focuses on developing an automated system for crop disease detection that integrates advanced computer vision techniques with machine learning algorithms. The system is designed to be user-friendly, providing farmers with a powerful tool that can be easily deployed in the field. This approach not only enhances the efficiency of disease management but also contributes to the sustainability of agricultural practices by reducing the need for chemical treatments and minimizing environmental impact.

TECHNOLOGICAL ACHIVEMENT

A. Early Developments

Initial efforts in crop disease detection using technology primarily focused on basic image analysis with manual feature extraction. Early systems were limited by computational power and heavily relied on human expertise for accurate diagnosis. Despite these constraints, these foundational systems paved the way for integrating technology into agriculture, demonstrating the potential for automated systems to enhance crop management.

B. Modern Innovations

With advancements in machine learning and computer vision, crop disease detection has evolved significantly. Modern systems leverage deep learning models, which can automatically extract and analyze features from images of crops. These models are trained on extensive datasets, allowing them to identify even subtle symptoms of various diseases with high accuracy. The integration of cloud computing and IoT devices enables real-time monitoring and analysis, providing farmers with immediate feedback on crop health.

C. Technological Components

Today's crop disease detection systems are characterized by their sophisticated algorithms and high-performance hardware. Key components include powerful GPUs for processing complex models, high-resolution cameras for capturing detailed images, and cloud-based platforms for data storage and analysis. These systems often incorporate web applications, allowing farmers to capture images of crops in the field and receive instant diagnostics and recommendations. The combination of these technologies results in a robust and scalable solution that can be adapted to various agricultural environments, improving crop yield and sustainability.

IMPACT ON CUSTOMER EXPERIENCE AND EFFICIENCY

Our system is designed to deliver exceptional performance in agricultural disease detection, leveraging the power of high-performance GPUs for processing complex deep learning models. Equipped with high-resolution cameras, the system captures detailed images of tomato plants, ensuring that even the smallest signs of disease are detected. At the core of our system is the ResNet backbone network, known for its deep architecture and superior feature extraction capabilities. To further enhance detection accuracy, we use k-means clustering to optimize the placement of anchors, ensuring that our model is finely tuned for identifying and localizing various tomato diseases. The incorporation of ROI (Region of Interest) pooling allows the system to focus on the most relevant parts of the image, increasing the precision of disease identification. Additionally, advanced classification methods are employed to distinguish between different types of diseases, making the system both robust and scalable. This comprehensive approach ensures that our system can be effectively deployed across diverse agricultural environments, providing farmers with a powerful tool to monitor plant health and take proactive measures against potential outbreaks.

MATERIALS AND MODULES

A. Tomato Disease Detection Method Based On Improved Faster R-CNN

The tomato disease detection model utilizing an improved Faster R-CNN is illustrated in Fig. 1. To enhance the model's effectiveness in disease detection, several modifications were made. First, ResNet was adopted in place of the VGG16 network. Tomato disease images of varying sizes are processed through ResNet to produce a deep feature map, which then serves as the input for both the Fast-RCNN and Region Proposal Network (RPN). Additionally, the k-means algorithm was employed to cluster the bounding boxes of tomato disease objects, optimizing the anchor points. In the RPN, this improved anchor framework is used to convolve the feature image, followed by bounding box (bbox) regression to locate the coordinates of the convolution outputs. These outputs are then classified using softmax to differentiate between foreground and background. Finally, the region of interest (ROI) is generated and used as input for Fast-RCNN. Within Fast-RCNN, the feature map is aligned with the candidate box coordinates provided by the ROIs, followed by max-pooling. The fully-connected layers and softmax are then utilized to classify the specific diseases within the region proposals, and bbox regression is employed to refine the position offsets, ultimately achieving a more accurate detection frame for tomato disease objects.

B. Feature Extraction Network

The feature extraction process aims to derive the original image features. Initially, images are normalized to a consistent size, after which their features are extracted through convolutional pooling in the CNN network. This process generates a feature map that serves as input for the RPN and Fast-RCNN.

Typically, Faster R-CNN employs a VGG16 network for image feature extraction. However, ResNet, with its greater number of convolutional layers, can extract more detailed object features. ResNet also includes a skipping structure that allows it to bypass one or more layers, addressing the issue of gradient vanishing caused by layer stacking. The VGG16 network, being less capable of deeply extracting features for tomato leaf diseases, was replaced with ResNet in this study. The input image is first standardized before being fed into the feature extraction network for consistent output. During the feature extraction phase, the image size is reduced by 16 times, similar to the VGG16 process, with the MobileNet model used in the experimental section also following this approach.

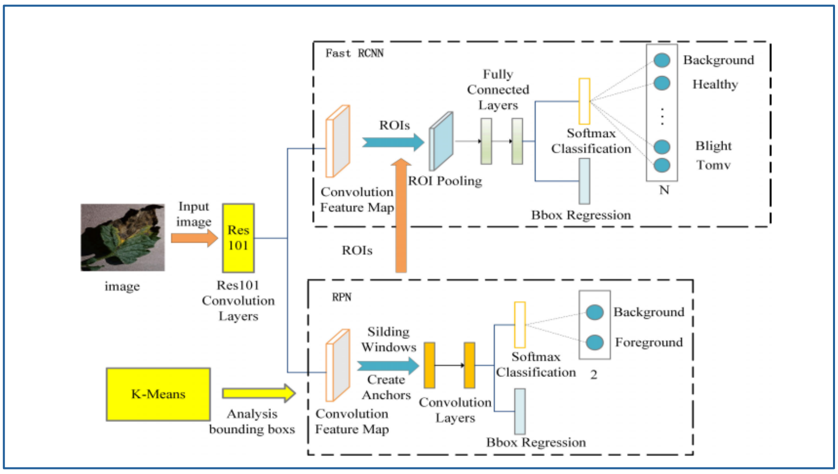

Fig.1: Tomato disease leaf detection model based on improved Faster RCNN

The deep residual network, implemented as a feedforward neural network with skip connections, enables the output of one layer to be passed directly across several layers as input to a subsequent layer. This method avoids the introduction of additional coefficients and does not significantly increase computation. By performing cross-layer operations and reusing intermediate features, the network mitigates gradient vanishing issues that arise with an increased number of neural network layers. Consequently, the deep residual network is highly effective in recognition tasks and is also widely applied in fault diagnosis and fault-tolerant control.

C. Ground Truth Generation

To have a robust and accurate training procedure, it is essential to determine the affected portion from the suspected samples correctly. To accomplish this, we have used LabelImg41, an open-source software to develop plant images' annotations to locate the RoIs. The final output containing the coordinate values of affected regions is saved in a CSV file which is later passed along with the input images to the Faster-RCNN framework for model training. Some examples of annotated samples are exhibited in Fig. 2

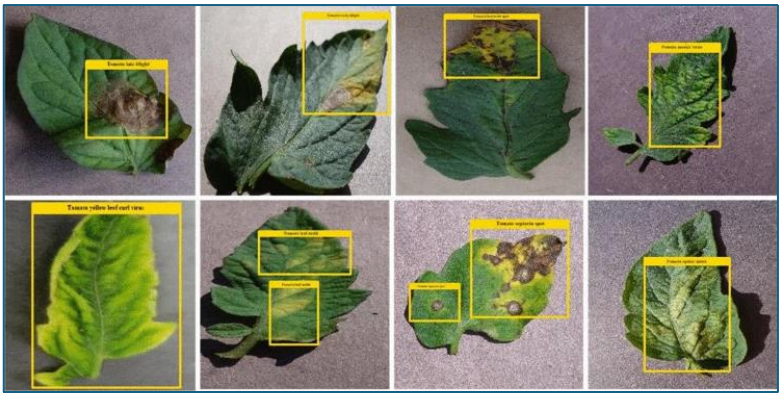

Fig.2: Annotated image samples

D. Region Proposal Network

In the Faster R-CNN framework, the Region Proposal Network (RPN) is a crucial component that connects with the feature extraction network. It is central to the Faster R-CNN model, tasked with positioning tomato disease leaves. RPN generates region proposals using a sliding window technique, centred on each point of the convolutional feature map output by the last shared convolutional layer. This sliding window moves across each point of the feature map to generate k anchor points. As a result, the bbox regression layer produces 4K outputs indicating the position coordinates of the tomato diseased leaves, while the classification layer produces 2K outputs to determine whether the tomato leaves are diseased (foreground) or not (background). Given that the classification layer has only two definitive outputs, we illustrate two deterministic branches in Fig. 1.

Faster R-CNN typically utilizes nine anchors computed from three scales (128^2, 256^2, 512^2) and three aspect ratios (1:1, 1:2, 2:1). These anchors cover most objects in object detection, but they are not optimally configured for our tomato disease leaf dataset. To enhance the anchor sizes, we introduced the k-means algorithm, which allows the anchors to align more closely with the bounding box dimensions in our dataset.

K-means clustering is a classic unsupervised learning algorithm that classifies data without requiring labelled inputs. The algorithm measures similarity between data objects based on distance, grouping similar objects into the same category.

In our network framework, k-means operates independently of the feature extraction network. It is not directly applied to the RPN but is used to determine the scale and size of the bounding boxes by clustering the dataset’s bounding boxes. The resulting bounding box sizes are then input into the RPN to enhance the anchor points.

The clustering process for tomato disease bounding boxes via k-means involves the following steps:

1. The bounding box sizes of tomato disease samples are used as input for the k-means algorithm. Assuming there are n bounding box samples, a value k is set to define the number of clusters.

2. k is randomly selected from the tomato sample as cluster points. The distance between each disease boundary box sample and its cluster point is calculated. Each sample is then assigned to the cluster point with the smallest distance, completing a clustering iteration.
3. The cluster points’ positions are recalculated based on the bounding box samples grouped by each cluster point.
4. A threshold t is established, and the distance between the new and original cluster points is calculated. If the distance exceeds t, steps 2 and 3 are repeated. If the distance is less than t, the result stabilizes and converges, yielding the final clustering of the tomato disease leaf bounding boxes.

E. Roi Pooling and Classification Regression

Based on the mapping relationship between the feature maps and the original image, the bounding boxes are transformed into corresponding regions of interest (ROIs) on the feature maps. Max-pooling is used to obtain a fixed-size tomato disease feature map and to complete the normalization process. The ROI pooling layer significantly accelerates tomato disease diagnosis and enables end-to-end training.

The classification task is carried out using softmax. After the fully-connected operation, the feature map is converted into a feature vector, which softmax uses to calculate the probability of each tomato disease. The regression task is performed in parallel with the classification task. The regression process refines the bounding box of the diseased leaves, ensuring that the experimentally generated bounding box of the tomato disease aligns closely with the actual object position.

RESULTS AND DISCUSSION

In this section, we present the experimental dataset, evaluation methods, and discuss the results from three key perspectives. First, we analyzed the bounding boxes of tomato disease images using the k-means clustering algorithm to enhance the anchor boxes. Second, we evaluated different feature extraction networks for analyzing tomato disease leaves. Lastly, we conducted a comparative analysis of these networks combined with the improved anchors.

A. Image Datasets

The training dataset was sourced from Kaggle, which consists of laboratory data. This dataset includes 61 categories. We used a Python script to process the annotation files, which contain the category numbers for each disease. By mapping these numbers to the corresponding image names, we isolated images of tomato diseases. Four specific tomato disease categories were selected for the experiment, resulting in a dataset of 4,178 images.

The dataset was randomly split into three subsets: 60% for training, 30% for validation, and 10% for testing. Figure 2 displays sample images representing healthy tomato leaves and four disease categories (powdery mildew, blight, leaf mold fungus, and ToMV). Each leaf exhibits distinct characteristics, such as varying textures, color brightness, and shading effects. Image annotation was performed using the Label Image tool .

B. Experimental Environment and Algorithm Evaluation

The introduced work was implemented in Python language with Tensorflow and Keras libraries and executed on an Nvidia RTX 3050 GPU-based system in Windows 11 environment.

The Faster-RCNN approach comprises four steps: feature extraction module, Region proposal networks (RPN), RoI pooling, and classification to locate and classify several plant diseases.

TP (true positive) represents the number of positive samples classified correctly in the prediction result. FP (false positive) represents the number of negative samples for which the prediction result is incorrectly detected. FN (false negative) represents the number of undetected positive samples in the prediction results defined in Eqs. (2) and (3) and Fig.3 respectively.

$$\mathbf{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+FP+TN+FN}} \qquad (2)$$

$$\mathbf{mAP} := \sum_{i=1}^{T} \mathbf{AP}(\mathbf{t_i})/\mathbf{T} \qquad (3)$$

In Eq. (3), AP denotes the average precision of each class and t is the query or test image. T is the total number of test samples.

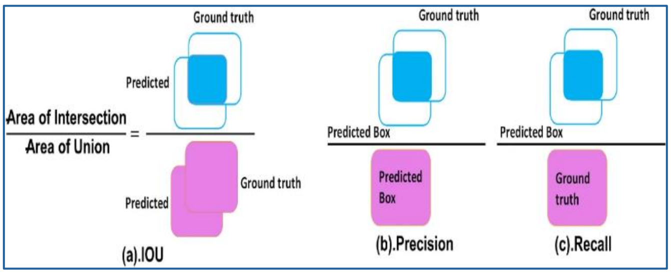

Fig.3: Visual demonstration of IOU, precision, and recall

C. Localization Results

The main characteristic of an automated plant disease system is that it should be capable of localizing the various categories of plant diseases accurately. Therefore, to evaluate the detection performance of the introduced solution, the tomato images from the employed database and show some of the visual results on Fig.4.

Fig.4: Localization results

D. Classification Results

To design an accurate and robust tomato plant leaf disease classification model, it should be able to differentiate between the various classes of plant abnormalities. For this reason, we have designed an experiment to discuss the classification results of our approach in detail. In this section, we have demonstrated the category-wise identification and classification results of ResNet-34-based Faster-RCNN in the form of precision, recall, F1-score, and an error rate.To evaluate the classification performance of the proposed ResNet-34-based Faster-RCNN on ten classes, we have drawn the boxplots (Figs. 5,6) to show the obtained precision and recall rates as these plots have the ability to better show the results by exhibiting the minimum, maximum, median values along with the symmetry and skewness of the data.

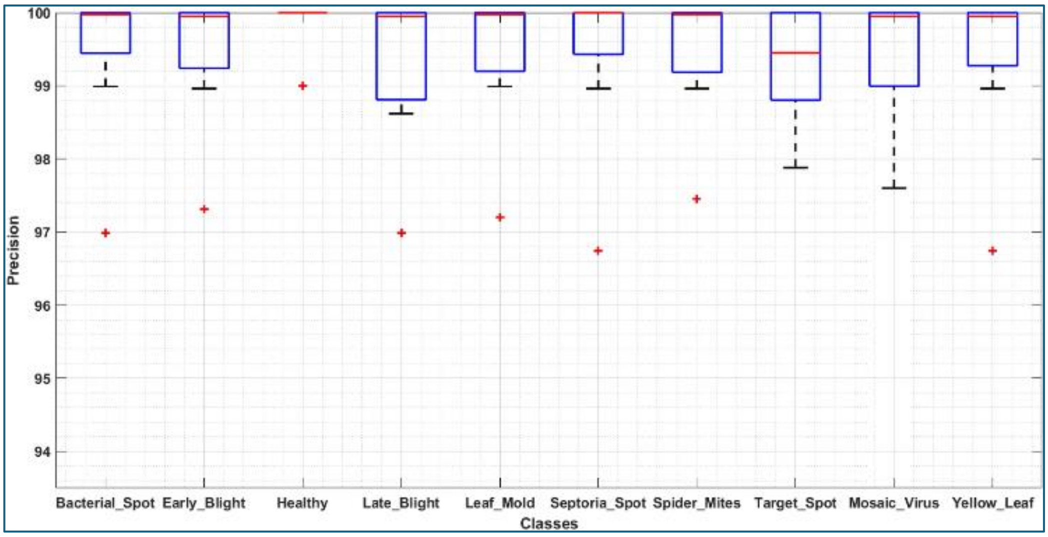

Fig.5: Evaluation using Precision.

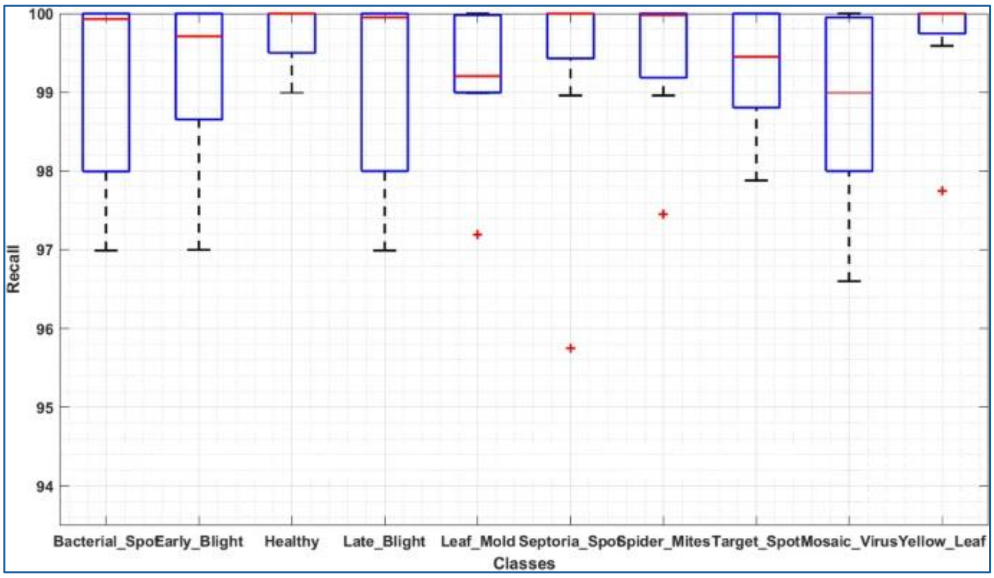

FIG.6: Proposed method evaluation using Recall.

Furthermore, the F1-score and the error rate for the entire dataset is shown in Fig.7.More clearly, we have obtained an average F1-score of 99.42% along with the minimum and maximum error rates of 0% and 0.35%, respectively.

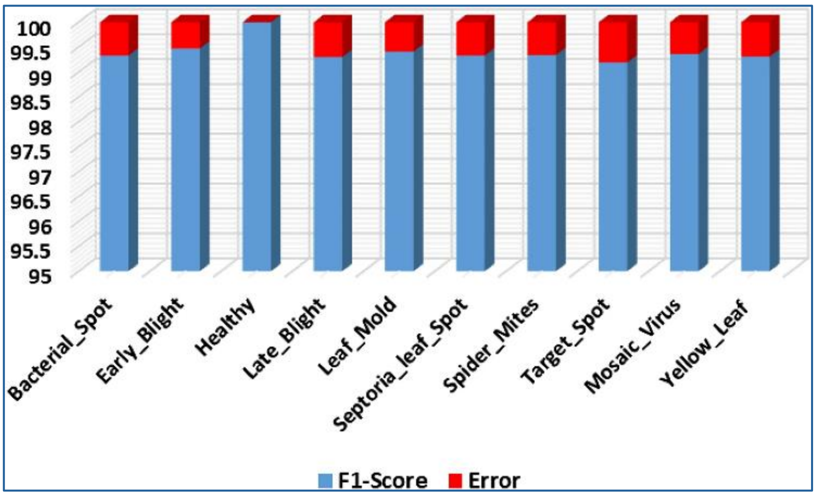

Fig.7: Evaluation using F1 score and error rate.

It is quite evident from the reported results that our approach has better recall ability and can better generalize to tomato plant diseases of several types.

CONCLUSION

We enhanced the Faster RCNN algorithm to effectively detect and recognize diseased tomato leaves, addressing both disease identification and leaf localization. By incorporating the k-means algorithm to optimize anchor boxes, and replacing VGG16 with ResNet101 for deeper feature extraction, our approach demonstrated improved accuracy compared to the original Faster RCNN. However, our current dataset is limited to laboratory conditions, allowing for the detection of only single leaf diseases in each image. Additionally, tomato diseases often manifest in fruits and stems, and future research will integrate these factors for a more comprehensive diagnostic approach. This expansion will contribute significantly to advancing smart agriculture by providing a more holistic plant disease detection system. Future work will focus on expanding this approach to include images of natural plants, capturing the complexity of real-world scenarios

.

ACKNOWLEDGMENT

The authors also gratefully acknowledge the insightful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

1. Z. Iqbal, M. A. Khan, M. Sharif, J. H. Shah, M. H. ur Rehman, and K. Javed, ‘‘An automated detection and classification of citrus plant diseases using image processing techniques: A review,’’ Comput. Electron. Agricult., vol. 153, pp. 12–32, Oct. 2018.
2. Y. Saijo, E. P.-I. Loo, and S. Yasuda, ‘‘Pattern recognition receptors and signaling in plant-microbe interactions,’’ Plant J., vol. 93, no. 4, pp. 592–613, Feb. 2018.
3. T. Pridmore, S. Tsaftaris, and H. Scharr, ‘‘Computer vision problems in plant phenotyping,’’ in Proc. CVPPP, 2018, pp. 1–4.
4. S. Cubero, N. Aleixos, E. Moltó, J. Gómez-Sanchis, and J. Blasco, ‘‘Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables,’’ Food Bioprocess Technol., vol. 4, no. 4, pp. 487–504, May 2011.

5. L. Fei, G. Lu, W. Jia, S. Teng, and D. Zhang, ‘‘Feature extraction methods for palmprint recognition: A survey and evaluation,’’ IEEE Trans. Syst., Man, Cybern. Syst., vol. 49, no. 2, pp. 346–363, Feb. 2019.
6. M. Abadi, ‘‘Tensorflow: A system for large-scale machine learning,’’ in Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI), 2016, pp. 265–283.
7. S. D. Khirade and A. B. Patil, ‘‘Plant disease detection using image processing,’’ in Proc. Int. Conf. Comput. Commun. Control Autom., Feb. 2015, pp. 768–771.
8. Y. Sasaki, T. Okamoto, K. IMOU, and T. TORII, ‘‘Automatic diagnosis of plant disease,’’ J. Jpn. Soc. Agricult. Mach., vol. 61, no. 2, pp. 119–126, 1999.
9. C. Dong, C. C. Loy, and X. Tang, ‘‘Accelerating the super-resolution convolutional neural network,’’ in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 391–407.
10. M. El-Helly, S. El-Beltagy, and A. Rafea, ‘‘Image analysis based interface for diagnostic expert systems,’’ in Proc. Winter Int. Symp. Inf. Commun. Technol., Dublin, Ireland, 2004, pp. 1–6.
11. M. Sammany and T. Medhat, ‘‘Dimensionality reduction using rough set approach for two neural networks-based applications,’’ in Proc. Int. Conf. Rough Sets Intell. Syst. Paradigms. Cham, Switzerland: Springer, 2007, pp. 639–647.
12. T. Baum, A. Navarro-Quezada, W. Knogge, D. Douchkov, P. Schweizer, and U. Seiffert, ‘‘HyphArea—Automated analysis of spatiotemporal fungal patterns,’’ J. Plant Physiol., vol. 168, no. 1, pp. 72–78, 2011.
13. S. Ueda and M. Nakamura, ‘‘Edge detection device, image forming apparatus, and edge detection method,’’ Google Patents 10 106 353, Oct. 23, 2018.
14. Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ Nature, vol. 521, no. 7553, pp. 436–444, 2015.
15. E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, ‘‘Convolutional neural networks for large-scale remote-sensing image classification,’’ IEEE Trans. Geosci. Remote Sens., vol. 55, no. 2, pp. 645–657, Feb. 2017.
16. T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, ‘‘Feature pyramid networks for object detection,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2117–2125.
17. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘Imagenet classification with deep convolutional neural networks,’’ in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.
18. K. Simonyan and A. Zisserman, ‘‘Very deep convolutional networks for large-scale image recognition,’’ 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556
19. H. Lee and H. Kwon, ‘‘Going deeper with contextual CNN for hyperspectral image classification,’’ IEEE Trans. Image Process., vol. 26, no. 10, pp. 4843–4855, Oct. 2017.
20. H. Lee, S. Eum, and H. Kwon, ‘‘Cross-domain CNN for hyperspectral image classification,’’ in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Jul. 2018, pp. 3627–3630.
21. X. Sun, S. Mu, Y. Xu, Z. Cao, and T. Su, ‘‘Image recognition of tea leaf diseases based on convolutional neural network,’’ 2019, arXiv:1901.02694. [Online]. Available: http://arxiv.org/abs/1901.02694
22. S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster R-CNN: Towards real-time object detection with region proposal networks,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

23. K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
24. N. Dhanachandra, K. Manglem, and Y. J. Chanu, ‘‘Image segmentation using k-means clustering algorithm and subtractive clustering algorithm,’’ Procedia Comput. Sci., vol. 54, pp. 764–771, Jan. 2015.
25. F. Zang and J.-s. Zhang, Kirchhain, ‘‘Softmax discriminant classifier,’’ in Proc. 3rd Int. Conf. Multimedia Inf. Netw. Secur., Nov. 2011, pp. 16–19.
26. Y. Qin, S. He, Y. Zhao, and Y. Gong, ‘‘RoI pooling based fast multi-domain convolutional neural networks for visual tracking,’’ in Proc. 2nd Int. Conf. Artif. Intell. Ind. Eng. (AIIE), 2016, pp. 198–202.
27. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, ‘‘Generalized intersection over union: A metric and a loss for bounding box regression,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 658–666.
28. M. S. Fisher and N. Arab, ‘‘Apparatus, system, and method for image normalization using a Gaussian residual of fit selection criteria,’’ Google Patents 15 987 359, Sep. 20, 2018.
29. M. Andrychowicz, ‘‘Learning to learn by gradient descent by gradient descent,’’ in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 3981–3989.
30. Y. Wu, B. Jiang, and N. Lu, ‘‘A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices,’’ IEEE Trans. Syst., Man, Cybern. Syst., vol. 49, no. 10, pp. 2108–2118, Oct. 2019.
31. Y. Wu, B. Jiang, and Y. Wang, ‘‘Incipient winding fault detection and diagnosis for squirrel-cage induction motors equipped on CRH trains,’’ ISA Trans., to be published, doi: 10.1016/j.isatra.2019.09.020.