International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 11 | Nov 2026 www.irjet.net p-ISSN: 2395-0072

Coffee Leaf Disease Detection Using Image Processing and Deep Learning

Dhanush Gowda M[1], Chavan Kumar M H[2], Dr. Kavitha A S[3]

[1,2,3]Navkis College of Engineering, Hassan, 573217, Karnataka, India.
Approved by AICTE, New Delhi and Affiliated to Visvesvaraya Technological University, Belagavi, 590018, Karnataka

---

Abstract - *Leaf diseases take away roughly 20–30% of the coffee crop, and the usual way of dealing with them, an expert looking at the plant, is slow, costly and sometimes wrong. We built a tool that does a first check from a photograph. It sorts coffee leaf images into four classes: Healthy, Leaf Miner, Phoma and Rust. An uploaded image is first checked for a leaf, resized to 224 × 224 pixels and then classified by a MobileNetV2 Convolutional Neural Network (CNN). We trained the network in two stages using transfer learning. In the first the pre-trained base stayed frozen; in the second we fine-tuned part of it. The model sits behind a Flask web application, New Leaf Scan, which gives the predicted class, a confidence score, the probability of every class, a few intermediate image-processing views, an image-based infection percentage with a disease stage, and suggested remedies. It also saves a scan history and can produce PDF reports. So far we have only shown the pipeline working on sample leaf images through the web interface. Testing accuracy on a labelled test set comes next and is not part of this paper.*

Key Words: Coffee Leaf Disease, Image Processing, Convolutional Neural Network, Transfer Learning, Leaf Rust, Precision Agriculture.

1. INTRODUCTION

Coffee is grown in over 50 countries and is one of the most heavily traded crops in the world. India is a major producer. Its coffee grows in the shade of other trees, and a good part of it is exported. But coffee plants get sick easily. Diseases and pests are believed to take 20–30% of the harvest, so finding them accurately and early matters a great deal for precision agriculture [1].

The traditional approach is to ask an expert to look at the plant, or to send samples to a laboratory. That takes time and money, and small-scale farmers often cannot get either [2]. Luckily, the common diseases of coffee leaves are easy to see. Take coffee leaf rust (CLR). It begins with tiny, yellowish, oily spots on the top of the leaf. The spots spread into larger circles, turn orange or red, and end up brown with a yellow rim.

The first computer-vision studies worked with features designed by hand: colour, texture and shape. These were passed to classifiers such as Support Vector Machines (SVM), K-Nearest Neighbors or decision trees [3], [4], [5]. Designing the features is hard work, and the results suffer when the lighting, background or leaf angle changes. CNNs largely removed this problem. Their convolutional layers learn the features themselves, from simple edges up to complicated lesion patterns [6], [7], [8]. Transfer learning helps further. By starting from a model already trained on a large image collection, we need less data and less training time.

In this paper we describe a CNN-based system for finding disease on coffee leaves. It takes a leaf image, preprocesses it in several steps, extracts features and classifies the image with the network, and then shows the prediction in a web interface. We are not trying to replace laboratory diagnosis. The aim is a fast, cheap first screening tool for farmers, nurseries, research institutions and extension workers, especially in places where expert help is scarce.

2. LITERATURE SURVEY

Many researchers have looked at detecting plant diseases with image processing and machine learning. Shrivastava and Pradhan [9] classified rice diseases using the colour features of leaf images, with the aim of depending less on experts. Han et al. [10] proposed a computer-vision method that detects crop diseases and also judges how severe they are. Kaur et al. [3] combined fractional-order Zernike moments with an SVM to recognise diseases on grape leaves.

Some studies deal with coffee itself. Marcos et al. [11] used a genetic algorithm to detect coffee leaf rust. Mettleq and Abu-Naser [12] and Suhartono et al. [13] built rule-based expert systems, which propose a likely diagnosis from symptoms that

the user types in. They are useful when no plant pathologist is around, but they depend on the user describing the problem rather than showing an image.

Deep learning has been used too. Durmus et al. [14] applied deep networks to tomato leaf disease, and Arnal Barbedo [15] studied how to identify a disease from single lesions and spots. Krohling et al. [16] released BRACOL, a dataset of Brazilian Arabica coffee leaf images labelled with diseases and pests. Shijie et al. [17] showed that augmenting the data improves CNN classification when there are few images. Naidu and Kumaravelan [18] and Ganatra and Patel [19] wrote reviews of machine-learning methods for plant leaf disease detection and classification.

Some gaps are still open. Methods based on hand-picked features can miss complex disease patterns. Deep models need enough varied data to train on. And very few systems combine image processing, a trained model and an easy interface that explains the result to people who are not specialists. We try to close these gaps by putting a trained CNN inside a web application for coffee leaves and testing it on a coffee leaf dataset with several classes.

3. DATASETS

A model can only be as good as its training data, and the variety in that data matters as well. Our classifier was trained on coffee leaf images kept in Train, validation and test folders. Each folder has four sub-folders, one per class: Healthy, Miner, Phoma and Rust. The images are all RGB photographs of coffee leaves.



Fig -1: Sample coffee leaf images of the four classes: Healthy, Leaf Miner, Phoma and Rust.

The classes are easy to tell apart by eye. A healthy leaf is evenly green. Rust shows up as yellow-orange powdery lesions. Phoma leaves have dark patches of dead tissue. Leaf Miner damage looks like pale, winding tunnels and blotches, made by larvae feeding inside the leaf.

The images were split into training, validation and test subsets, and the test set was never used during training. Augmentation was applied to the training images only: random rotation up to 25°, horizontal and vertical shifts up to 15%, zoom up to 20%, shear up to 15%, horizontal flips, and brightness changes between 0.8 and 1.2 times [17]. The classes differ in size, so we calculated balanced class weights from the training labels and applied them while training. Validation and test images were only normalised.

4. METHODOLOGY

4.1 Overall Pipeline

The system runs in a fixed order: image acquisition, preprocessing, feature extraction, model training and prediction output. Figure 2 gives the overview. It starts when the user uploads an image and ends when the predicted disease is displayed.

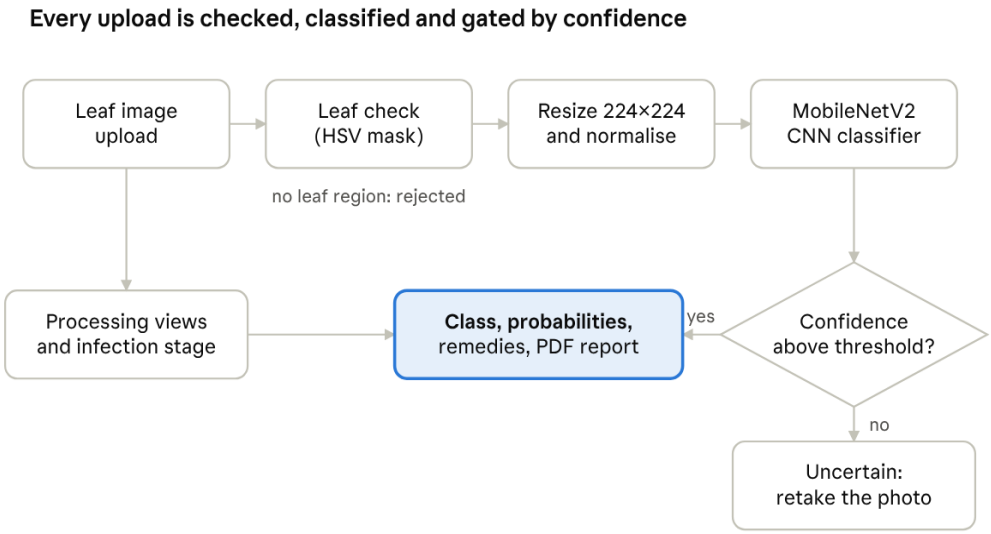

Fig -2: Overall process flow of the New Leaf Scan system.

4.2 Preprocessing

Leaf photographs differ in noise, lighting and background, and these differences can reduce the model's accuracy. So before an image reaches the network, it goes through several preprocessing steps that clean and normalise it.

An uploaded image is first read with OpenCV and converted to RGB. Next comes a leaf check. It keeps the pixels inside a green-to-brown HSV range and cleans up the mask with morphological closing and opening. If the largest region is smaller than 2% of the frame, the image is rejected, which stops photographs without a leaf from being classified. An accepted image is resized to 224 × 224 pixels and normalised with the MobileNetV2 preprocessing function, which scales pixel values to between −1 and 1.

The system also creates five views that help users look at the lesions. These are a grayscale image; a Canny edge map (thresholds 100 and 200), which brings out sharp changes in intensity at the leaf edge and in the lesions; an Otsu-thresholded binary image, which separates bright areas from dark ones; a sharpened image made with a 3 × 3 sharpening kernel; and a boundary overlay that draws the leaf outline found from the HSV leaf mask.

4.3 Feature Extraction

Classical methods calculate features by hand, such as colour histograms, texture descriptors like the Gray-Level Co-occurrence Matrix (GLCM), and shape measures. We do not. In our network the convolutional layers learn what is useful during training. The early layers pick up edges and colour changes, and the deeper layers respond to textures and shapes specific to lesions. No separate feature engineering is needed.

4.4 Deep Learning Model

The core of the system is a CNN that puts each leaf image into one of the four classes. Its convolutional layers build feature maps from learned kernels, ReLU activations make the model non-linear, and batch normalisation keeps training stable. We used MobileNetV2 pre-trained on ImageNet as the base, without its original top layers. On top of it come global average pooling, a fully connected layer of 256 ReLU units, a dropout layer with rate 0.40, and a softmax layer with four units, which gives one probability per class. The most probable class is reported as the prediction.

Transfer learning made the model more accurate and shortened training. We trained in two stages. In Stage 1 the MobileNetV2 base stayed frozen and only the new classification head was trained, with the Adam optimizer at a learning rate of 0.001 for up to 20 epochs. In Stage 2 we unfroze the base to fine-tune it, but left the first 100 layers and all batch-normalisation layers frozen, and trained on at a learning rate of 1e-5 for up to 30 epochs. Both stages used categorical cross-

entropy loss, a batch size of 32 and balanced class weights. The best weights were saved according to validation accuracy. To prevent overfitting, early stopping (patience of 7 and 8 epochs) restored those best weights, and the learning rate was multiplied by 0.3 each time validation loss stopped improving. The model was written in Python with TensorFlow/Keras.
To judge performance we use accuracy, precision, recall and F1-score. Together they show how often the predictions are right, and whether false positives and false negatives are balanced across the classes.

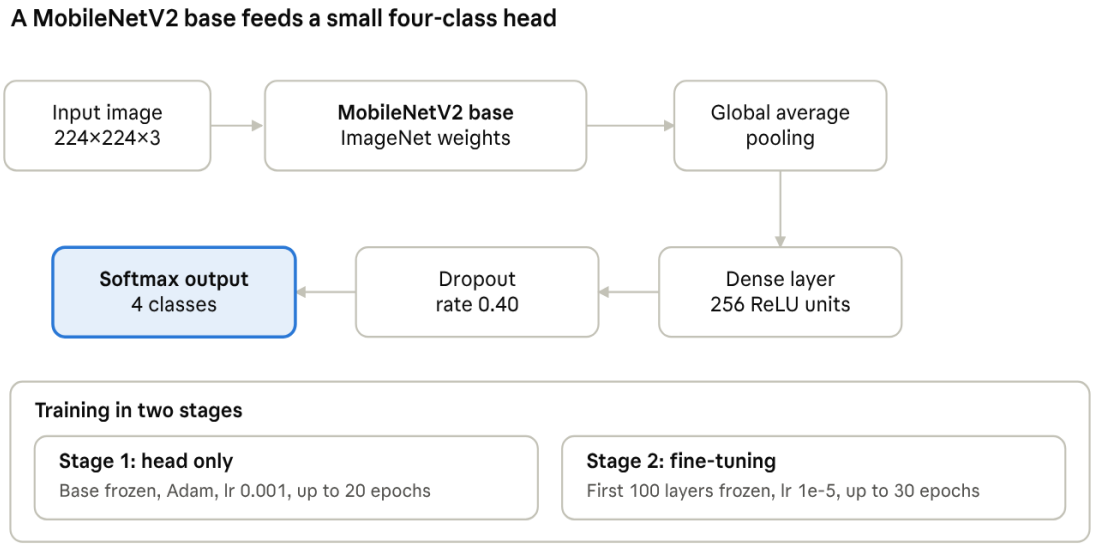

Fig -3: MobileNetV2-based classifier architecture and two-stage training.

4.5 System Architecture and Web Interface

The trained model runs inside a Flask web application called New Leaf Scan, which is written in Python with TensorFlow, OpenCV and a SQLite database. To use it, a person uploads a leaf image, either by dragging it onto the page or with the Choose File button (JPG or PNG, up to 5 MB), checks the preview, and clicks Analyze Image. The interface is available in English and Kannada, and has a dark theme and an Expert View.

On the backend, the image goes through the leaf check, preprocessing and the model. The application sends back the predicted disease, the model's confidence, the probability for each of the four classes, and the five image-processing views. A prediction is accepted only if its confidence reaches the threshold for that class, which is 70% for Healthy and 80% for Miner, Phoma and Rust. If it does not, the result is shown as Uncertain and the user is advised to retake the photograph. The leaf mask is also used to calculate an image-based infection percentage, a leaf health score, and the length and width of the leaf in pixels. For leaves predicted as diseased, the stage is Early below 20% infection, Moderate below 50% and Advanced at 50% or above. These numbers come from image analysis, not from laboratory measurement. Each class has its own remedy suggestions. Every scan is stored in a history with a dashboard. The application also offers a downloadable PDF report (ReportLab), a voice read-out of the result, and a coffee-disease chatbot.

5. RESULTS AND DISCUSSION

We tested the trained model through the New Leaf Scan interface using sample coffee leaf images. This section describes what the system displays for a sample. We do not give a quantitative evaluation on a labelled test set (accuracy, precision, recall and F1-score) in this paper. It is listed as future work.

The interface shows the prediction with a confidence bar and a breakdown over all classes. For a healthy leaf sample, for instance, the system answered Healthy with 79.20% model confidence and gave the rest of the probability to Rust (16.60%), Phoma (3.51%) and Leaf Miner (0.70%) (Figure 4). It also shows the processing stages of the uploaded image, and for diseased predictions it gives remedy suggestions such as field monitoring, traps and copper-based fungicides.

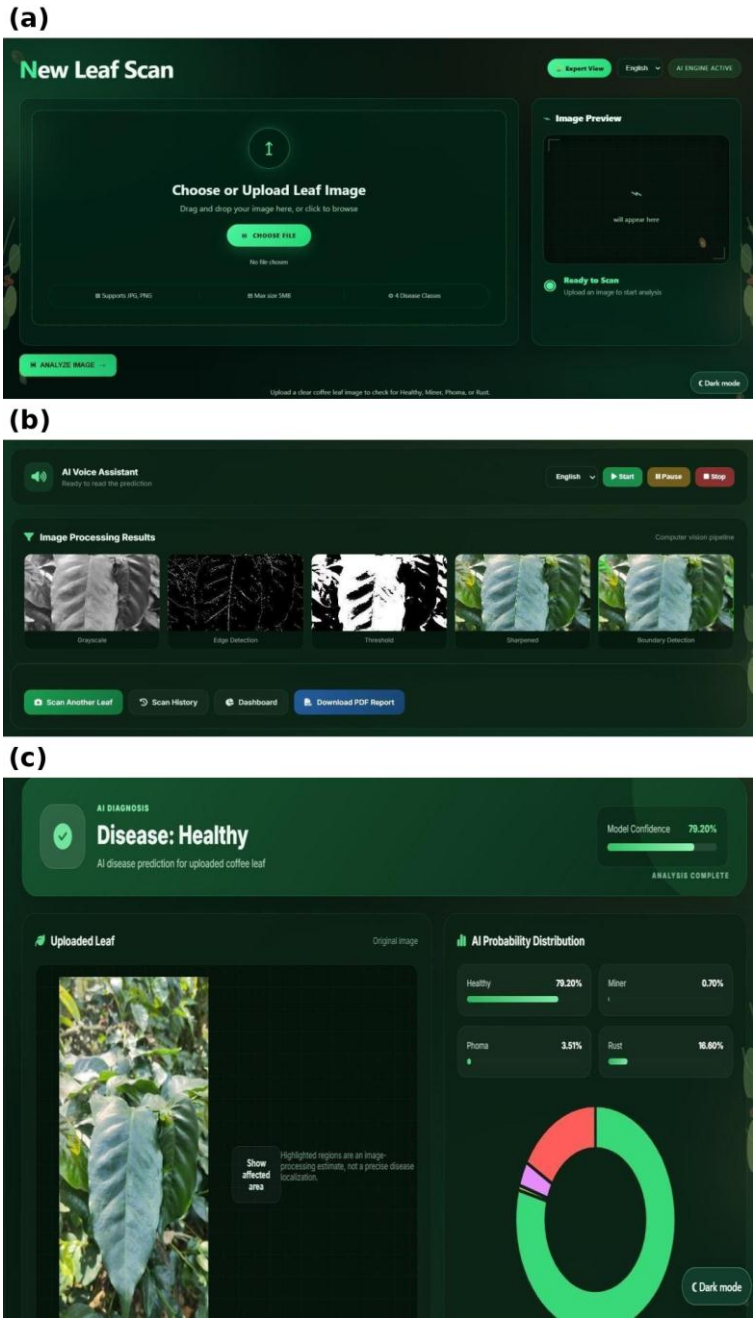

Fig -4: New Leaf Scan web interface: (a) upload screen, (b) image-processing views (grayscale, edge detection, threshold, sharpened, boundary), (c) prediction result for a healthy leaf with the class probability distribution.

Figure 5 shows the accuracy and loss recorded during training. Over the 12 epochs plotted, training accuracy rose from about 62% to about 94% and training loss fell from about 0.96 to about 0.17. Validation accuracy stayed much lower, between about 48% and 70%, and validation loss stayed between about 0.88 and 1.26 without a steady decline. The wide gap between the training and validation curves suggests that the model is overfitting to the training images, or that the validation images differ from them. Either way, the model generalises less well than its training accuracy suggests.

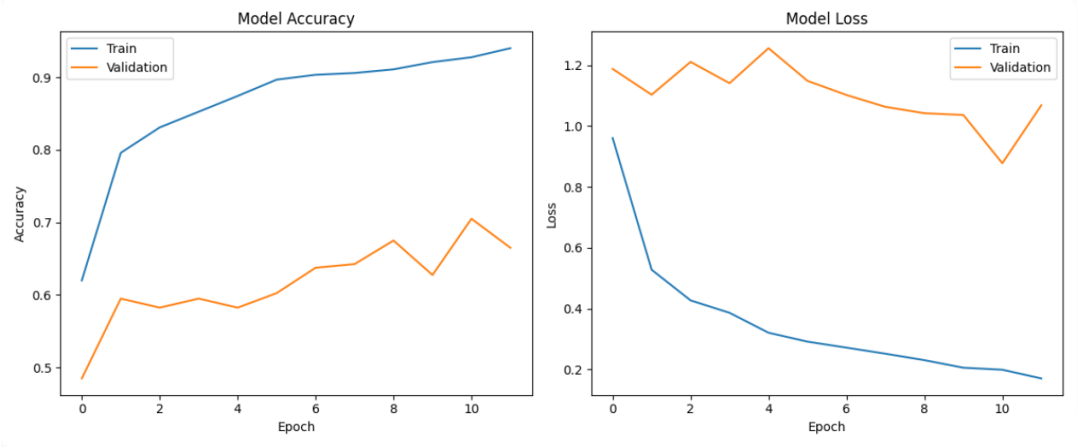

Fig -5: Model accuracy and loss on the training and validation sets during training.

There is one point to keep in mind. The softmax confidence only tells us how sure the model is when choosing among the classes it knows. It is not the real probability that the answer is right. A result of about 79% should be treated as a first indication, and users should ask an agricultural expert to confirm it before they treat the plants.

6. LIMITATIONS

The main limitation is simple: this paper does not evaluate the classifier on a held-out test set, so we claim no accuracy, precision, recall or F1-score. The confidence values in the interface belong to single predictions and are not a measure of overall accuracy.

The training curves in Figure 5 also point to a problem. Validation accuracy stays well below training accuracy, so the model has not yet been shown to work reliably on images it has not seen. Reducing this gap, for example with more varied training data, stronger regularisation or a closer look at how the validation set was built, is needed before any performance claim can be made.

There are other limitations we should be honest about. First, the dataset may be small, and it may not cover the full variety of coffee types, growth stages, lighting conditions and cameras. Deep learning usually needs thousands of samples per class.

Second, image quality matters a lot. Blurred, dark, partly hidden or low-resolution photographs can confuse the preprocessing and lower accuracy, and busy field backgrounds make segmentation harder.

Third, different diseases can look similar, and one leaf can have more than one problem. Right now the system gives only one class per image and does not estimate severity.

Last, the system assists with prediction but is not a laboratory diagnosis. An agronomist should check its remedy suggestions, and data protection and responsibility for a wrong diagnosis have to be thought through before deployment.

7. CONCLUSION

We have described a full system for detecting coffee leaf disease using image processing and deep learning. It joins a multi-stage preprocessing pipeline to a CNN classifier built with transfer learning, which puts each leaf image into a disease class. A web interface then delivers the result, the class probabilities and the remedies to non-expert users in near real time.

We showed the interface working from start to finish on sample leaf images. It gave the predicted class, the class probabilities, the processing views, the infection indicators and the remedy suggestions. The main contribution is a practical, low-cost pipeline for screening coffee leaves for disease from images. It is not a substitute for an expert diagnosis, but it can support early detection where specialists are hard to reach. The most immediate next step is to evaluate the classifier on a held-out test set.

In future work we plan to enlarge and balance the dataset using field images, add more diseases and severity estimation, and try transformer and attention-based models. We would also like to run lightweight models offline on smartphones and drones, combine weather and soil data to predict outbreaks, and link the system to IoT-based farm monitoring.

REFERENCES

1) L. S. P. Annabel, T. Annapoorani and P. Deepalakshmi, "Machine learning for plant leaf disease detection and classification," Int. Conf. on Communication and Signal Processing, Apr. 2019.

2) S. A. Miller, F. D. Beed and C. L. Harmon, "Plant disease diagnostic capabilities and networks," Annual Review of Phytopathology, vol. 47, pp. 15–38, 2009.

3) P. Kaur, H. S. Pannu and A. K. Malhi, "Plant disease recognition using fractional-order Zernike moments and SVM classifier," Neural Computing and Applications.

4) V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," Information Processing in Agriculture, vol. 4, no. 1.

5) G. Dhingra, V. Kumar and H. D. Joshi, "Study of digital image processing techniques for leaf disease detection and classification," Multimedia Tools and Applications, vol. 77, no. 15.

6) R. Sujatha, J. M. Chatterjee, N. Z. Jhanjhi and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," 2021, art. 103615.

7) L. Li, S. Zhang and B. Wang, "Plant disease detection and classification by deep learning," IEEE Access, 2021.

8) M. E. H. Chowdhury, T. Rahman, A. U. Khan and A. Khandakar, "Automatic and reliable leaf disease detection using deep learning techniques," AgriEngineering, vol. 3, no. 2, pp. 294–312, 2021, doi:10.3390/agriengineering3020020.

9) V. K. Shrivastava and M. K. Pradhan, "Rice plant disease classification using color features," doi:10.1007/s42161-020-00683-3, Oct. 2020.

10) L. Han, M. S. Haleem and M. Taylor, "A novel computer vision-based approach to automatic detection and severity assessment of crop diseases," Science and Information Conf. (SAI), London, 2015, pp. 638–644.

11) P. Marcos, N. L. Silva Rodovalho and A. R. Backes, "Coffee leaf rust detection using genetic algorithm," XV Workshop de Visão Computacional (WVC), São Bernardo do Campo, Brazil, 2019, pp. 16–20.

12) A. S. A. Mettleq and S. S. Abu-Naser, "A rule based system for the diagnosis of coffee diseases," 2019.

13) D. Suhartono, W. Aditya, M. Lestari and M. Yasin, "Expert system in detecting coffee plant diseases," Int. J. Electrical Energy, 2013, pp. 156–162, doi:10.12720/ijoee.1.3.156-162.

14) H. Durmus, E. O. Gunes and M. Kirci, "Disease detection on the leaves of the tomato plants by using deep learning," 6th Int. Conf. on Agro-Geoinformatics, 2017.

15) J. G. Arnal Barbedo, "Plant disease identification from individual lesions and spots using deep learning," Biosystems Engineering, vol. 180, pp. 96–107, 2019.

16) R. A. Krohling, G. J. M. Esgario and J. A. Ventura, "BRACOL – A Brazilian Arabica coffee leaf images dataset to identification and quantification of coffee diseases and pests," Mendeley Data, v1, 2019, doi:10.17632/yy2k5y8mxg.1.

17) J. Shijie, W. Ping, J. Peiyi and H. Siping, "Research on data augmentation for image classification based on convolution networks," Chinese Automation Congress (CAC), Jinan, 2017, pp. 4165–4170.

18) M. V. A. Naidu and G. Kumaravelan, "Machine learning approaches in plant leaf disease detection and classification," Proc. ICICV, IEEE, 2021.

19) N. Ganatra and A. Patel, "A multiclass plant leaf disease detection using image processing and machine learning techniques," Int. J. Emerging Technologies, 2020.