


Plant Disease Classification Based on ConvLSTM U-Net with Fully Connected Convolutional Layers



Meshal Alharbi¹, Suresh Kumar Rajagopal², Surendran Rajendran^{3*}, Mohammed Alshahrani⁴

¹ Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Alkharj 11942, Saudi Arabia

² Center for System Design, Chennai Institute of Technology, Chennai 600069, India

³ Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602105, India

⁴ Department of Mathematics, College of Sciences and Humanities, Prince Sattam Bin Abdulaziz University, Alkharj 11942, Saudi Arabia

Corresponding Author Email: surendranr.sse@saveetha.com

<https://doi.org/10.18280/ts.400114>

ABSTRACT

Received: 13 December 2022

Accepted: 10 February 2023

Keywords:

plant disease detection, ConvLSTM, U-Net, convolutional neural network, convolutional layers

Plants are susceptible to a variety of illnesses throughout their growth stages. One of the trickiest issues in agriculture is the early diagnosis of plant diseases. The entire output may be negatively impacted by infections if they are not discovered early on, which would lower farmers' profitability. Numerous researchers have proposed numerous cutting-edge solutions based on Deep Learning and Machine Learning techniques to address this issue. However, the majority of these systems either has poor classification accuracy rates or utilizes millions of training parameters. In this research, a novel model using ConvLSTM U-Net-based automatic detection of plant disease is proposed. To the best of our knowledge, no state-of-the-art systems described in the literature have a hybrid system based on CAE and CNN to automatically identify plant diseases. The proposed model employed in this study is to identify the presence of Bacterial Spot disease in medicinal plants using the image of their leaves, but it may be extended to identifying any plant disease. The work conducted for this research employ a dataset that is readily accessible to get images of medicinal plant leaves. In comparison to previous methods described in the literature, the proposed ConLSTM U-Net model requires for less training parameters. As a consequence of this, the amount of time necessary to train the model for automatic plant disease detection and the amount of time required to diagnose the disease in plants using the trained model are both significantly decreased.

1. INTRODUCTION

The ability to automatically identify plant diseases from leaf samples is a significant step forward in agriculture. In addition, the prompt and accurate diagnosis of plant diseases has a major influence on both crop yield and quality [1]. Due to the wide variety of crops grown, even a trained agronomist or pathologist may miss a plant disease's telltale symptoms on the leaves. However, in rural sections of impoverished countries, visual examination is still the major technique of disease diagnosis [2]. Experts must keep a constant eye on it as well. Farmers in more remote areas would have to spend time and money travelling to consult with an expert [3, 4]. Automated computational methods for plant disease identification and diagnosis are useful for farmers and agronomists due to their high throughput and accuracy.

Deep learning is an emerging technique in machine learning (ML) that is being used in many different areas of study [5]. Deep learning enables the direct use of raw data [6] without the need for the usage of manually produced features. The use of deep learning in computer vision has received significant attention in recent years, leading to the development of a number of new approaches in the field [7]. The CNN-based

system proposed by Lu et al. [8] can accurately identify 10 common rice diseases, including rice blast, rice false smut, rice sheath blight, foolish seedling disease, rice bacterial leaf blight, rice brown spot, rice seedling blight, rice sheath rot, rice bacterial sheath root, and rice bacterial wilt [9].

Many methods for assisting farmers and agricultural professionals in diagnosing illnesses have been identified using advanced image processing and pattern recognition algorithms. Using images and other artificial intelligence-based technologies, the quality of agricultural and aquaculture items may also be automatically rated [10]. To construct a plant disease detection system, photographs of different plant sections may be obtained. The most common site to locate plant ailments is on a plant's leaves. Even though image processing technologies are useful in recognizing plant illnesses, these systems are prone to disparities in leaf photographs owing to changes in shape, color, texture, and other characteristics. Models for deep learning and machine learning may be trained using these photographs. In recent years, various deep learning techniques have been applied to the world of agriculture to handle a number of challenges, including insect identification, fruit detection, and classification of plant leaves, fruit disease detection, and leaf

disease detection. Real-time implementation of traditional machine learning algorithms for illness detection is fairly tough. Deep learning methods may therefore pave the way to overcoming these obstacles and establishing expert systems that will assist the agriculture business. Plant diseases have been detected and recognized using a number of approaches. The bulk of plant disease symptoms may be revealed by a very complex analysis of images of plant leaves. Due to the diversity of crops and the complexity of psychopathological problems, even trained agronomists often miss signs of plant sickness. Using deep learning and computer vision-assisted techniques, field experts and farmers may better diagnose plant diseases through the analysis of input leaf images.

Numerous approaches have been developed by researchers to address the aforementioned problems. A wide variety of feature sets may be used by machine learning for plant disease classification. Among these feature sets, the traditional hand-crafted ones and the ones based on deep learning (DL) are the most well-liked. Preprocessing, including image improvement, color alteration, and segmentation [11], is required prior to efficient feature extraction. The next step after feature extraction is to apply a classifier. Some examples of common classifiers are K nearest neighbors [12], support vector machine [13], random forest, decision tree [14], logistic regression, rule generation, naive bayes [15], Deep CNN, and artificial neural networks. Using similarity measurements (such as distance, proximity, or closeness) to address classification issues, KNN is a simple supervised machine learning approach [16]. A second popular supervised machine learning technique for classification, SVM [17, 18], has gained a lot of attention recently. The purpose of SVM [19, 20] is to identify a hyperplane that divides the data into distinct groups. In order to provide forecasts, NB classifiers use many probability metrics [21]. It's based on the assumption that the qualities that were artificially introduced have no causal relationship with one another [22]. An ANN is a kind of network with an output that is modelled after the neurons in the human brain [23]. The network is composed of an input layer, a processing layer, and an output layer. Learning is achieved by adjusting the weights [24]. Strong classification results may be obtained using handmade feature-based techniques, but these methods aren't without their limitations, such as the need for expensive and time-consuming preprocessing. The handcrafted-based approach has limited feature extraction, and the recovered attributes may not be enough for effective identification, so the accuracy goes down.

Machine learning and other statistical approaches perform poorly since they depend on manual characteristics for operation. As a result, techniques based on deep learning were developed to identify various plant diseases in large datasets. The leaves of the *Vigna mungo* plant may be classified as healthy, moderate, or severe using a convolutional neural network [25]. When training the sequential network on images, several different preprocessing strategies are used. The overall accuracy of the model using images from different categories was 97.403%. The Efficient Net architecture [26] uses transfer learning to train a model using laboratory images of several plant leaf diseases for disease classification [27]. In the alpine steppes of northern Tibet, hyperspectral imaging is also used to identify plant species. Differentiating between plant species in challenging environments with high spatial homogeneity has been accomplished using principal component analysis, spectral indices, continuum reduction, and derivatives [28]. A total of 94.73 percent used four different types of machine

learning approaches. Using imaging and convolutional neural network methods, researchers were able to combat bacteriosis, a common ailment of peach harvest [29]. A variety of adaptive strategies for determining the optimal color image channel and grey-level slicing for analyzing leaf photos. Bacteriosis is detected with 98.75 percent accuracy by the deep learning model. The PD2 SE-Net is an AI-assisted network developed for diagnosing plant diseases and quantifying their impact [30]. A total of five crops were divided into three groups for the study, and the Resnet-50 architecture was utilized to train a variety of images. Another study using transfer learning to diagnose cassava plant disease found it to be 93% accurate when presented with unlabeled images [31]. All the positive findings from the studies that have been published so far don't negate the need for further research into how to develop AI-based systems with the sensitivity and specificity needed to accurately identify plant species and categories and detect illnesses. For these automatic classification frameworks to be more reliable and useful, they should be trained on a large number of crops in different classes and imaging settings.

The remainder of the paper is organized into four subsections. In Section 2 of this study, some cutting-edge technologies for automated plant disease detection are explored. The construction of the suggested hybrid model and the processes involved are discussed in Section 3. Section 4 of this research presents the model's findings for identifying peach plants infected with bacterial spots. Finally, Section 5 concludes the paper.

2. MATERIALS AND METHODS OF THE PROPOSED MODEL

Image processing is a technique for extracting quantitative information from images proposed in Ensemble Classifier for Plant Disease Detection (ECPDD) [32]. These image processing methods have been applied to real-world issues in a wide variety of fields, including medical imaging, remote sensing, robotic vision, pattern recognition, video processing, and color processing, and so on. In agriculture, these methods have been useful for a wide variety of tasks, including estimating crop yields, assessing the quality of fruits and vegetables, and diagnosing illnesses in leaves. The quality of plants and crops is greatly diminished by the overuse of fertilizers and pesticides; professionals can typically detect plant illnesses merely by looking at them. This work proposes an ensemble model for plant disease detection at the leaf level, using Random Forest and K-Nearest Neighbor (KNN). Images of plant diseases were utilized as the benchmark dataset. One thousand photographs of both healthy and diseased leaves (brown rust, early blight, and late blight) are included in this collection.

In Automatic Recognition of Medicinal Plants using Machine Learning Techniques (ARMP), a wide range of individuals may benefit from improved plant species identification [33]. Among them are foresters, taxonomists, botanists, pharmaceutical labs, doctors, organizations working to save endangered species, governments, and the general public. As a result, there is a growing need for machine-based plant identification capabilities. It has been proven to be possible to fully automate the identification of medicinal plants using computer vision and machine learning approaches. Twenty-four different medicinal plant species had their leaves picked in a controlled environment and shot using a smart

phone. Each leaf's length, breadth, area, perimeter, colour, number of vertices, and hull area were recorded. Finally, based on these primary features, a large number of derivative features were calculated. Using a random forest classifier and a 10-fold cross-validation technique yielded the most promising results. With an accuracy of 90.1%, the random forest classifier outperforms the k-nearest neighbour, support vector machine, naive Bayes, and neural network methods.

This fundamental rationale provides the basis for new state-of-the-art segmentation and classification algorithms [34], and is a major driving force in the Particle Swarm Optimization with fuzzy C means based segmentation and machine learning classifier for leaf disease prediction (PSOFCM). This paper's primary focus is on developing a unique approach to disease prediction in leaves. This study examined a novel and effective method for segmenting images, extracting features, and classifying plant leaf diseases. The proposed technique begins with image preprocessing for plant leaves, then applies background removal using the Gaussian Mixture Model (GMM), followed by segmentation using fuzzy c-means with assistance from Particle Swarm Optimization (PSO) (PSOFCM). Calculating vein and form features, edge-based feature extraction, and texture features (TF). In order to classify medicinal plant leaves, this method employs the Multiple Kernel Parallel Support Vector Machine (MK-PSVM) classifier.

In the Classification of Medicinal Plant Leaves Based on Multispectral and Texture Features Using Machine Learning Approach (MTFML), the Department of Agriculture at Islamia University of Bahawalpur in Pakistan gathered the leaves of six different types of medicinal plants for the dataset. Scientifically, these plants are referred to by their Latin names: *Mentha balsamea*, *Ocimum sanctum*, *Melissa officinalis*, *Aegle marmelos*, *Stevia rebaudiana*, and *Nepeta cataria*. (Herbal) Common English names for these plants include Peppermint, Tulsi, Lemon, balm, Bael, Stevia, and Catnip [35]. Computer vision lab space is required to collect the multispectral and digital image datasets. They scale down the leaf and convert it to grayscale before they begin working on it. The second stage is to apply a Sobel filter to detect edges and lines in the data based on the strength of the seeds. There are a total of 65 fused features retrieved, all of which are a mashup of texture, run-length matrix, and multispectral features. To start the feature optimization process, they had selected 14 primary features using a chi-square feature selection strategy. 14 optimized features are Texture Energy Average, Correlation Range, Inverse Diff Range, Texture Entropy Range, 45dgr_GLevNonU, Vertl_GLevNonU, S (5, 5) Entropy, Skewness, 135dgr_RLNonUni, R, G, B, NIR, and SWIR.

The primary contribution of the proposed study is the creation of a system for the identification and categorization of plant diseases based on deep learning. Based on their health and disease classification, plant leaves from four distinct crops have been taken into consideration [36]. The gathered dataset makes use of pictures from databases from various nations to ensure that the suggested framework is accepted globally. For creating a solid foundation, the photos include both laboratory and field images. Dense convolutional neural network architectures are trained on a big dataset of gathered pictures from diverse categories [37]. Images feature a lot of intra- and interclass variance and complicated backgrounds. For the purposes of studying and testing the framework, the collected dataset is separated into training, validation, and testing sets.

Real-time operation, resolution invariance, as well as the ability to link to camera systems for monitoring plant health are only some of the advantages of the proposed framework.

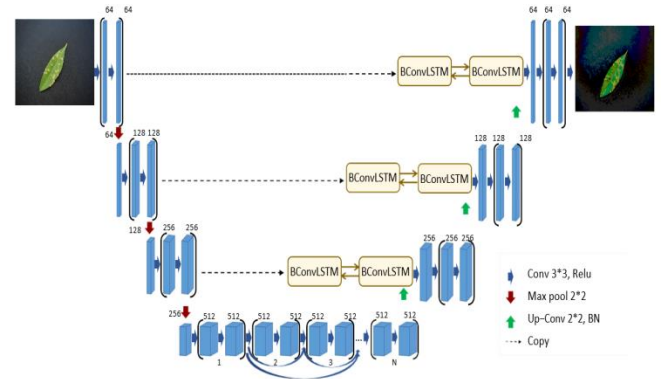


Figure 1. ConvLSTM U-Net with bidirectional connectivity and completely linked convolutional layers

There are four steps to the BCDU-Net contracting process. Convolutional 33 filters, a maximum pooling function of 22, and a ReLU are used in each step. Each stage increases the number of feature maps by a factor of two as depicted in Figure 1. Using a layer-by-layer expansion method, the contracting methodology constantly extracts picture representations. The last layer of encoding generates a high-dimensional visual representation with a significant amount of semantic information. A set of convolutional layers make up the last step of the original U-encoding Net's process. When a network includes a number of convolutional layers, the method learns many types of properties. Even yet, duplicate features from succeeding convolutions could be picked up by the network. This problem is addressed by densely coupled convolutions [38]. This helps the network operate better by using the idea of "collective knowledge," in which the feature maps are reused across the network. Before passing them on to be used as the input to the next convolution, it involves merging the feature maps learned from the current layer with the feature maps learned from all prior convolutional layers [39].

There are certain advantages to using densely coupled convolutions rather than standard convolutions. It helps the network learn new feature maps, rather than only relying on the same old ones. This idea also improves the network's representational power by facilitating the exchange of data between nodes and the recycling of previously used characteristics. To further mitigate the threat of gradient inflation or disappearance, densely coupled convolutions may get an advantage from all features produced before. Taking the opposite path to get to their proper nodes in the network also speeds up the transmission of gradients. The proposed network makes advantage of densely coupled convolutions. To do this, we add a single block in the form of two convolutions. Figure 2 depicts the last convolutional layer of the encoding procedure, which consists of a sequence of N blocks.

The output of the preceding layer must be upsampled before decoding can begin. Regular U-Nets have matching feature maps trimmed in the contracting route and then transferred to the decoding path. Following the up-sampling procedure, these feature maps are sent into the algorithm. To further refine the processing of these two feature maps, we use BConvLSTM in BCDU-Net. First, χd is sent to an up-convolutional layer (see Figure 3), where a 2 x 2 convolution

and an upsampling function are used to decrease the number of feature channels while simultaneously expanding the size of each feature map [40].

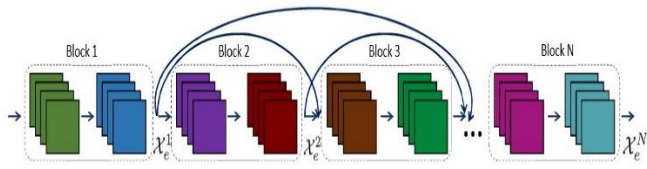


Figure 2. Dense Layer of Bi-Directional ConvLSTM U-Net

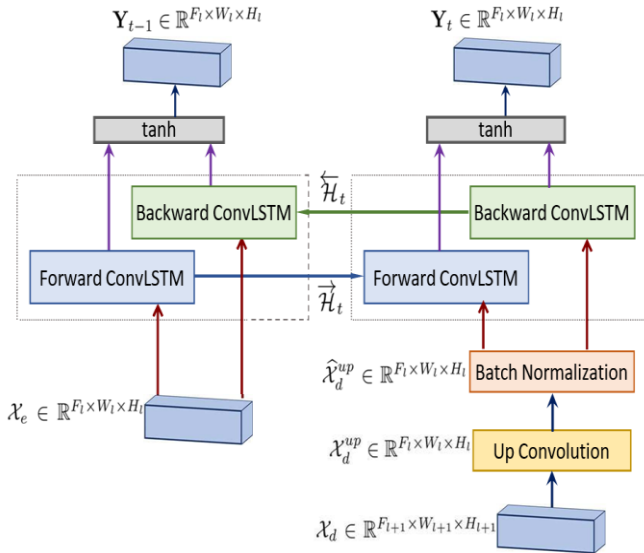


Figure 3. U-Net Bi-Directional ConvLSTM

$$i_t = \sigma(W_{xi} * \chi_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} * C_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xi} * \chi_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} * C_{t-1} + b_f) \quad (2)$$

$$C_t = f_t o C_{t-1} + i_t \tanh(W_{xc} * \chi_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo} * \chi_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} o C_t + b_o) \quad (4)$$

$$\mathcal{H}_t = o_t \tanh(C_t) \quad (5)$$

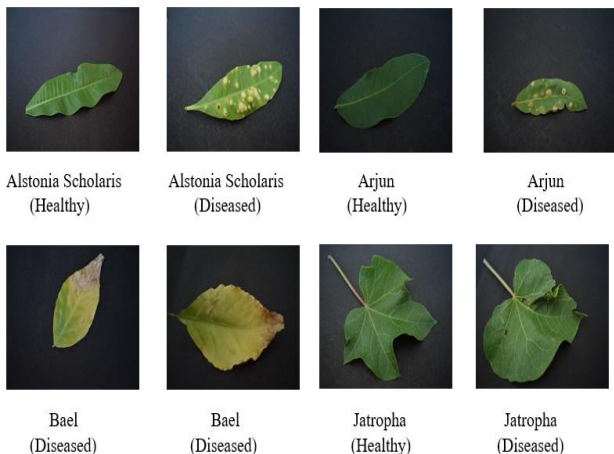


Figure 4. Leaf photos from several categories utilized in the suggested work

After splitting the input into forward and backward paths using two ConvLSTMs, BConvLSTM makes a judgement for the current input by taking into account the dependencies in both directions of the data. In a typical ConvLSTM, just the forward-direction dependencies are taken into account. However, the whole chain of information must be considered, therefore backward dependencies must be taken into consideration. Prediction accuracy was shown to increase by looking at data from both the present and the past. Standard ConvLSTMs may be thought of as both backward and forward ones. There are, therefore, two groups of settings to adjust for the forward and reverse states.

The disease known as bacterial spot has been identified in plants using the model that was presented. The Mendeley dataset has been mined for photos of plant leaves, and those results have been presented here. Arjun, Alstonia, Scholaris, Bael, and Jatropa are the four plants that have been chosen for this purpose. All four of these plants are good for both the economy and the environment. Images of these plants' leaf surfaces, both when they were healthy and when they were sick, have been collected and put into two different modules. The whole collection of pictures has been divided into two categories: healthy and ill. This was the primary method of organisation used. To begin with, the obtained photos are categorised and named in accordance with the plants. Figure 4 illustrates both a healthy leaf and a damaged leaf from a leaf plant, providing an example of each.

3. EXPERIMENTAL VALIDATION

Between the original U-Net and the new network that is being planned, there are many notable fundamental differences. Table 1 provides a summary of our findings on the "Accuracy" and "Average Processing time" of the original U-Net and its modifications for the 10 datasets that were utilised. After making any changes to the network, we evaluate each part for its effect on the final result. Table 1 depicts how the basic U-outcome Net's is enhanced by the addition of BConvLSTM to the skip connections. There are 10 samples of Alstonia Scholaris plant leaves from Mendeley dataset in Figure 5. The original U-Net and the BCDU-Net are shown with their percent accuracy for each sample. Compared with the original U-Net segmentation, the recommended network's results are more precise. In order to use the capabilities of the subsequent encoding layer and the preceding decoding layer, it is necessary to incorporate the skip connections. For the sake of shorthand, we refer to these as the encoded and decoded features. These two types of features are concatenated together in the initial version of U-Net.

In order to merge the information that had been encoded and decoded, we used a set of BConvLSTMs as a component of the proposed network. The encoded features are more detailed on a level that is measured in terms of individual pixels, yet the decoded qualities include a greater amount of semantic information. Due to the relative relevance of these two features, a collection of feature maps that are rich in both local and semantic information may be generated using them. Instead of a straightforward concatenation, we use the BConvLSTM algorithm, which allows us to combine the encoded and decoded features. For each feature class, the BConvLSTM algorithm applies a chain of convolution filters. This way, each ConvLSTM state that corresponds to a particular feature (such as an encoded feature) can encode relevant information about

a separate feature. Using convolutional filters and hyperbolic tangent functions, the network is able to learn advanced data architectures.

Table 2 and Figure 6 depict the accuracy in terms of percentage for the Arjun plant leaf with the comparison of existing and proposed methods. For the 100% data chunk, the accuracy is 96.84919%, which is much higher than the existing methods, ECPDD, ARMP, PSOFCM, and MTFML such as 94.99%, 89.75%, 89.79% and 95.09% respectively.

Table 3 and Figure 7 depict the accuracy in terms of percentage for the Arjun plant leaf with the comparison of existing and proposed methods. For the 100% data chunk, the accuracy is 98.79%, which is much higher than the existing methods, ECPDD, ARMP, PSOFCM, and MTFML, such as 96.94%, 91.69%, 91.74% and 97.04% respectively.

Table 4 and Figure 8 depict the accuracy in terms of percentage for the Arjun plant leaf with the comparison of existing and proposed methods. For the 100% data chunk, the accuracy is 98.79%, which is much higher than the existing methods, ECPDD, ARMP, PSOFCM, and MTFML, such as 96.94%, 91.69%, 91.74% and 97.04% respectively.

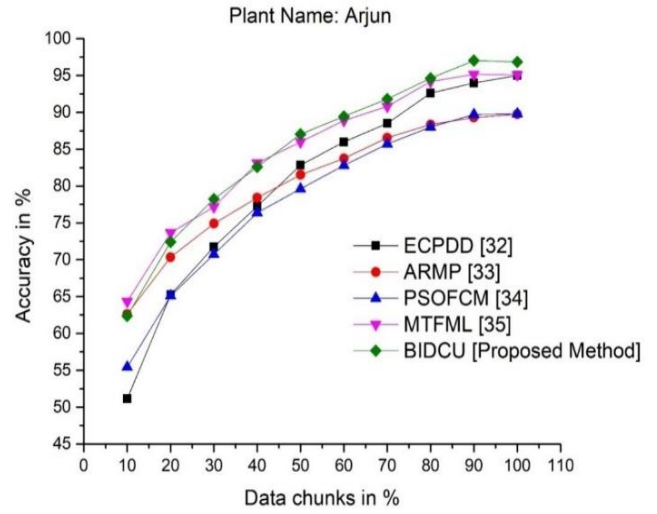


Figure 6. Accuracy for Arjun plant leaf

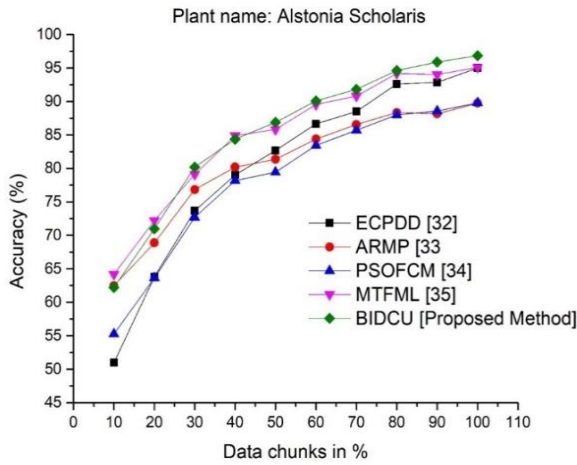


Figure 5. Accuracy for Alstonia Scholaris plant leaves

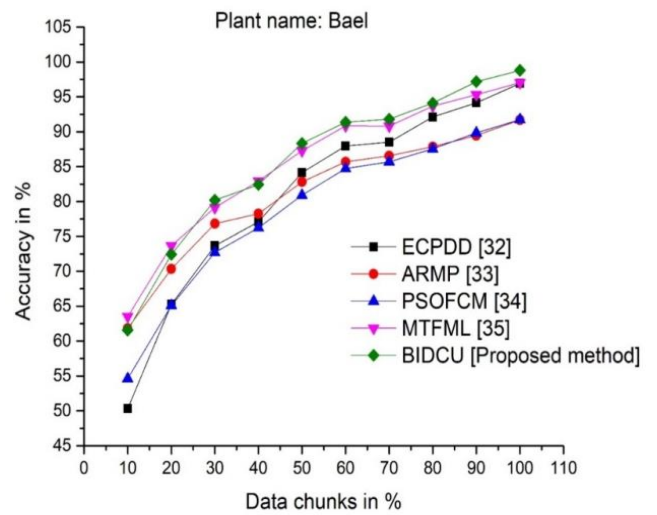


Figure 7. Accuracy for Bael plant leaf

Table 1. Parameters of accuracy (%): Plant name - Alstonia Scholaris

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU [Proposed Method]
10	50.97006	62.47006	55.27007	64.17006	62.22006
20	63.8101	68.89156	63.65083	72.21713	70.98042
30	73.67596	76.83777	72.68187	79.11423	80.19102
40	79.03364	80.19656	78.1651	84.89768	84.37428
50	82.68828	81.35683	79.44756	85.83125	86.86797
60	86.64868	84.39197	83.44533	89.54398	90.08407
70	88.49948	86.53772	85.6936	90.80124	91.80556
80	92.59455	88.33894	88.01674	94.16562	94.6055
90	92.85191	88.17555	88.56374	94.02846	95.88203
100	94.99919	89.74919	89.79919	95.09919	96.84919

Table 2. Parameters of accuracy (%): Plant name – Arjun

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU [Proposed Method]
10	51.13187	62.63187	55.43188	64.33188	62.38187
20	65.26641	70.34787	65.10715	73.67343	72.43674
30	71.73422	74.89603	70.74012	77.17249	78.24928
40	77.2537	78.41663	76.38516	83.11774	82.59434
50	82.85009	81.51863	79.60936	85.99307	87.02977
60	86.00143	83.74472	82.79808	88.89673	89.43682
70	88.49948	86.53772	85.6936	90.80124	91.80556
80	92.59455	88.33894	88.01675	94.16562	94.60551
90	93.9846	89.30824	89.69642	95.16114	97.01473
100	94.99919	89.74919	89.79919	95.09919	96.84919

Table 3. Parameters of accuracy (%): Plant name – Bael

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU [Proposed Method]
10	50.32281	61.82282	54.62282	63.52281	61.57282
20	65.26641	70.34787	65.10715	73.67343	72.43674
30	73.67596	76.83777	72.68187	79.11423	80.19102
40	77.09189	78.25481	76.22335	82.95593	82.43253
50	84.14459	82.81313	80.90385	87.28757	88.32427
60	87.94318	85.68646	84.73982	90.83848	91.37856
70	88.49948	86.53772	85.6936	90.80124	91.80556
80	92.10912	87.8535	87.5313	93.68018	94.12007
90	94.14642	89.47006	89.85824	95.32296	97.17654
100	96.94094	91.69094	91.74094	97.04094	98.79094

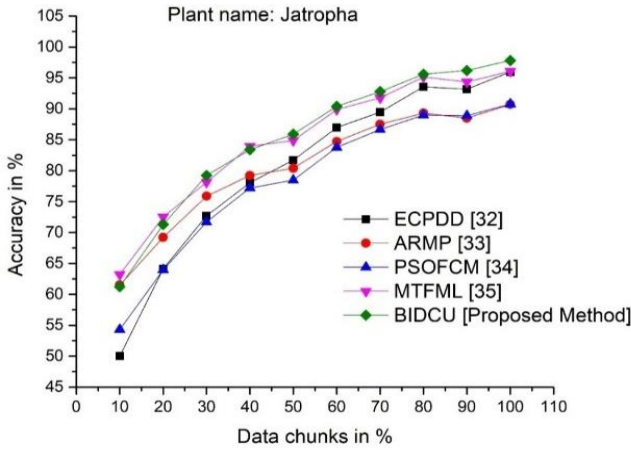


Figure 8. Accuracy for Jatropa plant leaves

Table 5 and Figure 9 depict the average processing time in terms of milliseconds for the Alstonia Scholaris plant leaves with the comparison of existing and proposed methods. For the 80% data chunk, the processing time is 1253ms, which is much lesser than the existing methods, ECPDD, ARMP, PSOFCM, and MTFML such as 1576ms, 1401ms, 1529ms and 1956ms respectively.

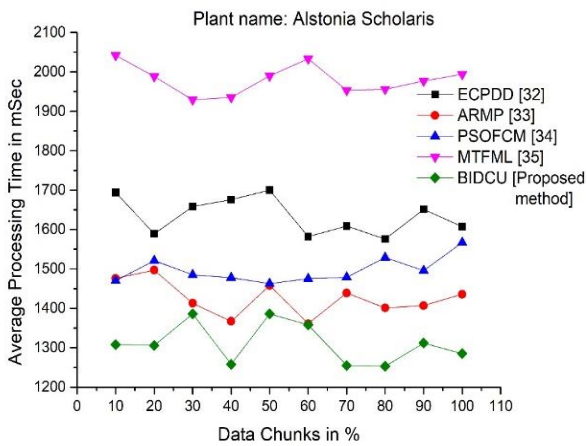


Figure 9. Average processing time (ms) for Alstonia Scholaris plant leaves

Table 6 and Figure 10 depict the average processing time in terms of milliseconds for the Arjun plant leaf with the comparison of existing and proposed methods. For the 100% data chunk, the processing time is 1273ms, which is much lesser than the existing methods, ECPDD, ARMP, PSOFCM, and MTFML such as 1595ms, 1424ms, 1555ms and 1982ms respectively.

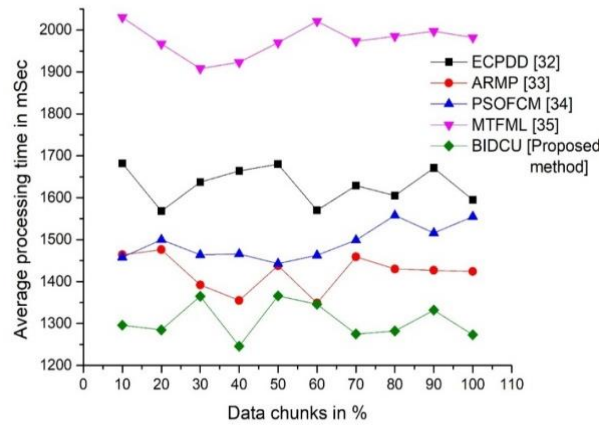


Figure 10. Average processing time (ms) for Arjun plant leaf

Table 7 and Figure 11 depict the average processing time in terms of milliseconds for the Bael plant leaf with the comparison of existing and proposed methods. For the 100% data chunk, the processing time is 1252ms, which is much lesser than the existing methods, ECPDD, ARMP, PSOFCM and MTFML such as 1574ms, 1403ms, 1534ms and 1961ms respectively.

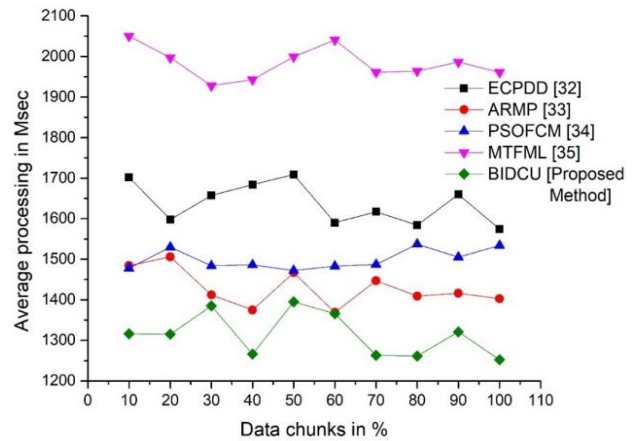


Figure 11. Average processing time (ms) for Bael plant leaves

Table 8 and Figure 12 depict the average processing time in terms of milliseconds for the Jatropa plant leaf with the comparison of existing and proposed methods. For the 100% data chunk, the processing time is 1257ms, which is much lesser than the existing methods, ECPDD, ARMP, PSOFCM and MTFML such as 1579ms, 1408ms, 1539ms and 1966ms respectively.

Table 4. Parameters of accuracy (%): Plant name – Jatropha

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU [Proposed Method]
10	49.99919	61.49919	54.29919	63.19919	61.24919
20	64.13373	69.21519	63.97446	72.54075	71.30405
30	72.70509	75.8669	71.711	78.14336	79.22015
40	78.06276	79.22569	77.19423	83.9268	83.4034
50	81.71741	80.38595	78.47668	84.86038	85.89709
60	86.97231	84.71558	83.76895	89.8676	90.40768
70	89.47035	87.50859	86.66447	91.77212	92.77643
80	93.56542	89.30981	88.98762	95.13649	95.57638
90	93.17554	88.49918	88.88736	94.35208	96.20566
100	95.97006	90.72006	90.77007	96.07006	97.82006

Table 5. Average processing time in millisecond: Plant name - Alstonia Scholaris

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU[Proposed Method]
10	1694	1476	1470	2042	1308
20	1589	1497	1521	1988	1306
30	1658	1413	1485	1929	1386
40	1676	1367	1478	1935	1258
50	1700	1458	1463	1990	1386
60	1582	1361	1475	2033	1358
70	1609	1439	1479	1953	1255
80	1576	1401	1529	1956	1253
90	1651	1407	1496	1977	1312
100	1607	1436	1567	1994	1285

Table 6. Average processing time in millisecond: Plant name – Arjun

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU[Proposed Method]
10	1682	1464	1458	2030	1296
20	1568	1476	1500	1967	1285
30	1637	1392	1464	1908	1365
40	1664	1355	1466	1923	1246
50	1680	1438	1443	1970	1366
60	1570	1349	1463	2021	1346
70	1629	1459	1499	1973	1275
80	1605	1430	1558	1985	1282
90	1671	1427	1516	1997	1332
100	1595	1424	1555	1982	1273

Table 7. Average processing time in millisecond: Plant name – Bael

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU[Proposed Method]
10	1702	1484	1478	2050	1316
20	1598	1506	1530	1997	1315
30	1657	1412	1484	1928	1385
40	1684	1375	1486	1943	1266
50	1709	1467	1472	1999	1395
60	1590	1369	1483	2041	1366
70	1617	1447	1487	1961	1263
80	1584	1409	1537	1964	1261
90	1660	1416	1505	1986	1321
100	1574	1403	1534	1961	1252

Table 8. Average processing time in millisecond: Plant name – Jatropha

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU [Proposed Method]
10	1666	1448	1442	2014	1280
20	1593	1501	1525	1992	1310
30	1662	1417	1489	1933	1390
40	1680	1371	1482	1939	1262
50	1705	1463	1468	1995	1391
60	1586	1365	1479	2037	1362
70	1613	1443	1483	1957	1259
80	1580	1405	1533	1960	1257
90	1655	1411	1500	1981	1316
100	1579	1408	1539	1966	1257

Table 9. True positive rate: Plant name - Alstonia Scholaris

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU [Proposed Method]
10	50.94126	62.09982	55.07421	63.61722	61.82283
20	63.85899	68.77007	63.65804	72.29578	71.09703
30	73.83021	76.87735	72.87568	79.15717	80.449
40	79.37925	80.24992	78.41412	84.78511	83.82508
50	83.90514	82.35631	80.76715	86.97337	87.73164
60	86.94219	84.49355	83.78128	89.46288	90.28326
70	88.31904	86.18639	85.59718	90.01264	91.90157
80	91.98917	87.79404	87.36592	93.32436	93.62748
90	93.13108	88.81493	88.50491	94.31529	96.27409
100	94.94611	89.11625	89.05016	94.16375	95.74307

Table 10. True negative rate: Plant name - Alstonia Scholaris

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU [Proposed Method]
10	51.00068	62.86368	55.48165	64.7697	62.64492
20	63.76156	69.01463	63.64364	72.13903	70.86509
30	73.5237	76.79831	72.4913	79.07142	79.93737
40	78.69607	80.14339	77.92041	85.01099	84.9416
50	81.55575	80.41724	78.23649	84.7576	86.04294
60	86.3598	84.29098	83.11598	89.6254	89.88683
70	88.68162	86.89593	85.79054	91.62155	91.70999
80	93.21764	88.89977	88.69064	95.04018	95.62838
90	92.57634	87.5569	88.62273	93.74532	95.49657
100	95.05241	90.40296	90.57752	96.07512	98.01013

Table 11. Parameter F-Measure (%): Plant name - Alstonia Scholaris

Data (%)	ECPDD [32]	ARMP [33]	PSOFCM [34]	MTFML [35]	BIDCU [Proposed Method]
10	51.70888	63.03559	56.11698	64.88291	62.84426
20	63.74616	68.99191	63.64124	72.16804	70.89999
30	73.5905	76.8207	72.56565	79.09885	80.10674
40	78.90958	80.17909	78.06901	84.92208	84.50011
50	82.37195	81.06436	78.99715	85.60899	86.71592
60	86.59543	84.36895	83.3626	89.5547	90.05948
70	88.52649	86.60275	85.71294	90.89101	91.79616
80	92.64755	88.42239	88.1202	94.22173	94.66531
90	92.82871	88.07735	88.57246	94.00907	95.86451
100	95.00214	89.83146	89.89609	95.15055	96.88683

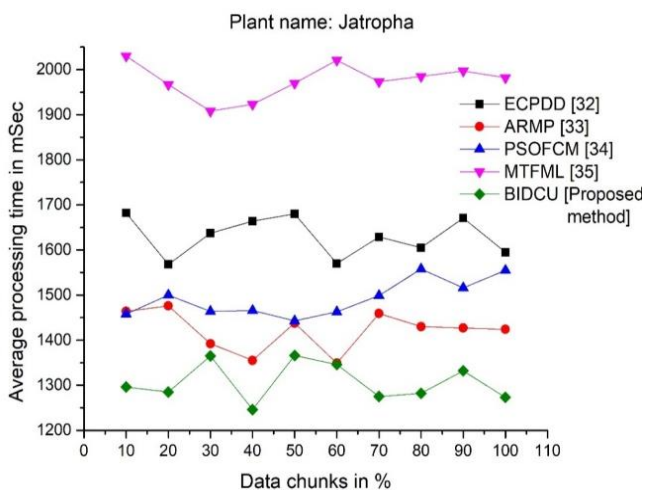
**Figure 12.** Average processing time (ms) for Jatropha plant leaves

Table 9, 10 and 11 depicts the true positive rate, true negative rate, and F-Measure for the plant leaf Alstonia Scholaris.

4. CONCLUSIONS

Utilizing digital imagery to identify and classify plant diseases is a difficult endeavour. Therefore, prompt detection of the plant disease is crucial for farmers and plant pathologists to take the appropriate response. The proposed methodology uses a total of four distinct types of plant leaf photos for this purpose. The images of the plants that were taken into consideration were from a variety of different categories and included both lab-view and live field photos of the plants. Inherent diversity was present in each of these photograph types. During training, the deep learning dense model is given a variety of photographs from a wide range of categories. To properly evaluate the model, this was then tested on the testing set's unseen photos. The proposed model demonstrated its applicability to identify plant diseases and classify them by achieving an average cross-validation accuracy of 85.39%, 85.24%, 85.82%, and 85.38% and an average processing time of 1310.7ms, 1306.6ms, 1314ms and 1308.4ms for the Alstonia Scholaris, Arjun, Bael and Jatropha plant leaf respectively. Other plant leaf photos will be taken into consideration in subsequent work in order to diversify the plant leaf dataset and aid the trained model in challenging conditions.

ACKNOWLEDGMENT

The authors would like to thank Prince Sattam Bin Abdulaziz University, Chennai Institute of Technology, and Saveetha School of Engineering for providing us with various resources and unconditional support for carrying out this study.

REFERENCES

- [1] Sethy, P.K., Barpanda, N.K., Rath, A.K., Behera, S.K. (2020). Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 175: 105527. <https://doi.org/10.1016/j.compag.2020.105527>
- [2] Chen, J., Chen, J., Zhang, D., Sun, Y., Nanekaran, Y.A. (2020). Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, 173: 105393. <https://doi.org/10.1016/j.compag.2020.105393>
- [3] Bai, X., Cao, Z., Zhao, L., Zhang, J., Lv, C., Li, C., Xie, J. (2018). Rice heading stage automatic observation by multi-classifier cascade based rice spike detection method. *Agricultural and Forest Meteorology*, 259: 260-270. <https://doi.org/10.1016/j.agrformet.2018.05.001>
- [4] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., Hughes, D.P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, 8: 1852. <https://doi.org/10.3389/fpls.2017.01852>
- [5] Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M., Alrahmaneh, Z. (2011). Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications*, 17(1): 31-38.
- [6] Mokhtar, U., Hassenian, A.E., Emary, E., Mahmoud, M.A. (2015). SVM-based detection of tomato leaves diseases. In *Advances In intelligent System and Computing*, pp. 641-652. https://doi.org/10.1007/978-3-319-11310-4_55
- [7] Rajagopal, S., Thanarajan, T., Alotaibi, Y., Alghamdi, S. (2022). Brain tumor: Hybrid feature extraction based on UNet and 3DCNN. *Computer Systems Science and Engineering*, 45(2): 2093-2109. <http://dx.doi.org/10.32604/csse.2023.032488>
- [8] Lu, Y., Yi, S., Zeng, N., Liu, Y., Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 267: 378-384. <https://doi.org/10.1016/j.neucom.2017.06.023>
- [9] Brahim, M., Boukhalfa, K., Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. *Applied Artificial Intelligence*, 31(4): 299-315. <https://doi.org/10.1080/08839514.2017.1315516>
- [10] Singh, A., Dutta, M.K., Jennane, R., Lespessailles, E. (2017). Classification of the trabecular bone structure of osteoporotic patients using machine vision. *Computers in Biology and Medicine*, 91: 148-158. <https://doi.org/10.1016/j.compbiomed.2017.10.011>
- [11] Camargo, A., Smith, J.S. (2009). An image-processing based algorithm to automatically identify plant disease visual symptoms. *Biosystems Engineering*, 102(1): 9-21. <https://doi.org/10.1016/j.biosystemseng.2008.09.030>
- [12] Singh, J., Kaur, H. (2019). Plant disease detection based on region-based segmentation and KNN classifier. In *Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*, pp. 1667-1675. https://doi.org/10.1007/978-3-030-00665-5_154
- [13] Vizhi, T., Varthini, P.B. (2016). Online vaccines and immunizations service based on resource management techniques in cloud computing. *Biomedical Research-India*, 27: S392-S399.
- [14] Chaudhary, A., Kolhe, S., Kamal, R. (2016). An improved random forest classifier for multi-class classification. *Information Processing in Agriculture*, 3(4): 215-222. <https://doi.org/10.1016/j.inpa.2016.08.002>
- [15] Phadikar, S., Sil, J., Das, A.K. (2013). Rice diseases classification using feature selection and rule generation techniques. *Computers and Electronics in Agriculture*, 90: 76-85. <https://doi.org/10.1016/j.compag.2012.11.001>
- [16] Munisami, T., Ramsurn, M., Kishnah, S., Pudaruth, S. (2015). Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers. *Procedia Computer Science*, 58: 740-747. <https://doi.org/10.1016/j.procs.2015.08.095>
- [17] Ebrahimi, M.A., Khoshtaghaza, M.H., Minaei, S., Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture*, 137: 52-58. <https://doi.org/10.1016/j.compag.2017.03.016>
- [18] Garcia-Ruiz, F., Sankaran, S., Maja, J.M., Lee, W.S., Rasmussen, J., Ehsani, R. (2013). Comparison of two aerial imaging platforms for identification of Huanglongbing-infected citrus trees. *Computers and Electronics in Agriculture*, 91: 106-115. <https://doi.org/10.1016/j.compag.2012.12.002>
- [19] Yao, Q., Guan, Z., Zhou, Y., Tang, J., Hu, Y., Yang, B. (2009). Application of support vector machine for detecting rice diseases using shape and color texture features. In *2009 international conference on engineering computation*, pp. 79-83. <https://doi.org/10.1109/ICEC.2009.73>
- [20] Surendran, R., Tamilvizhi, T. (2018). How to improve the resource utilization in cloud data center?. In *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 1-6. <https://doi.org/10.1109/3ICT.2018.8855740>
- [21] Tan, J.W., Chang, S.W., Kareem, S.B.A., Yap, H.J., Yong, K.T. (2018). Deep learning for plant species classification using leaf vein morphometric. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(1): 82-90. <https://doi.org/10.1109/TCBB.2018.2848653>
- [22] Duraisamy, K., Thanarajan, T., Alharbi, M. (2022). Implementation of omar pigeon space-time (OPST) algorithm to mitigate the interference and peak-to-average power ratio (PAPR) using RPR mobile and HST-HM in the 5G. *Traitement du Signal*, 39(5): 1631-1638. <http://dx.doi.org/10.18280/ts.390520>
- [23] Riya, K.S., Surendran, R., Tavera Romero, C.A., Sendil, M.S. (2023). Encryption with user authentication model for internet of medical things environment. *Intelligent Automation & Soft Computing*, 35(1): 507-520.
- [24] Pujari, D., Yakkundimath, R., Byadgi, A.S. (2016). SVM and ANN based classification of plant diseases using feature reduction technique. *IJIMAI*, 3(7): 6-14.
- [25] Joshi, R.C., Kaushik, M., Dutta, M.K., Srivastava, A.,

- Choudhary, N. (2021). VirLeafNet: Automatic analysis and viral disease diagnosis using deep-learning in Vigna mungo plant. *Ecological Informatics*, 61: 101197. <https://doi.org/10.1016/j.ecoinf.2020.101197>
- [26] Tan, M., Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105-6114.
- [27] Atila, Ü., Uçar, M., Akyol, K., Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics*, 61: 101182. <https://doi.org/10.1016/j.ecoinf.2020.101182>
- [28] Liu, E., Zhao, H., Zhang, S., He, J., Yang, X., Xiao, X. (2021). Identification of plant species in an alpine steppe of Northern Tibet using close-range hyperspectral imagery. *Ecological Informatics*, 61: 101213. <https://doi.org/10.1016/j.ecoinf.2021.101213>
- [29] Yadav, S., Sengar, N., Singh, A., Singh, A., Dutta, M.K. (2021). Identification of disease using deep learning and evaluation of bacteriosis in peach leaf. *Ecological Informatics*, 61: 101247. <https://doi.org/10.1016/j.ecoinf.2021.101247>
- [30] Liang, Q., Xiang, S., Hu, Y., Coppola, G., Zhang, D., Sun, W. (2019). PD2SE-Net: Computer-assisted plant disease diagnosis and severity estimation network. *Computers and Electronics in Agriculture*, 157: 518-529. <https://doi.org/10.1016/j.compag.2019.01.034>
- [31] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., Hughes, D.P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, 8: 1852. <https://doi.org/10.3389/fpls.2017.01852>
- [32] Yousuf, A., Khan, U. (2021). Ensemble classifier for plant disease detection. *International Journal of Computer Science and Mobile Computing*, 10(1).
- [33] Begue, A., Kowlessur, V., Singh, U., Mahomoodally, F., Pudaruth, S. (2017). Automatic recognition of medicinal plants using machine learning techniques. *International Journal of Advanced Computer Science and Applications*. 8(4): 166-75.
- [34] Sumithra, M.G., Saranya, N. (2021). Particle Swarm Optimization (PSO) with fuzzy c means (PSO-FCM)-based segmentation and machine learning classifier for leaf diseases prediction. *Concurrency and Computation: Practice and Experience*, 33(3): e5312. <https://doi.org/10.1002/cpe.5312>
- [35] Naeem, S., Ali, A., Chesneau, C., Tahir, M.H., Jamal, F., Sherwani, R.A.K., Ul Hassan, M. (2021). The classification of medicinal plant leaves based on multispectral and texture feature using machine learning approach. *Agronomy*, 11(2): 263. <https://doi.org/10.3390/agronomy11020263>
- [36] Tamilvizhi, T., Surendran, R., Romero, C.A.T., Sendil, M.S. (2022). Privacy preserving reliable data transmission in cluster based vehicular Adhoc networks. *Intelligent Automation & Soft Computing*, 34(2): 1265-1279. <https://doi.org/10.32604/iasc.2022.026331>
- [37] Song, H., Wang, W., Zhao, S., Shen, J., Lam, K.M. (2018). Pyramid dilated deeper ConvLSTM for video salient object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 715-731.
- [38] Tamilvizhi, T., Surendran, R., Anbazhagan, K., Rajkumar, K. (2022). Quantum behaved particle swarm optimization-based deep transfer learning model for sugarcane leaf disease detection and classification. *Mathematical Problems in Engineering*, 2022: 3452413. <https://doi.org/10.1155/2022/3452413>
- [39] Wagle, S.A., Ramachandran, H., Sampe, J., Mohammad, F., Md Ali, S.H. (2021). Effect of data augmentation in the classification and validation of tomato plant disease with deep learning methods. *Traitement du Signal*, 38(6): 1657-1670. <https://doi.org/10.18280/ts.380609>
- [40] Wagle, S.A., Ramachandran, H. (2021). A deep learning-based approach in classification and validation of tomato leaf disease. *Traitement du Signal*, 38(3): 699-709. <https://doi.org/10.18280/ts.380317>