

Department of Computer Science and Engineering
Bangladesh University of Business and Technology (BUBT)

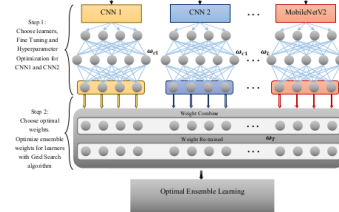


CSE 498: Literature Review Records

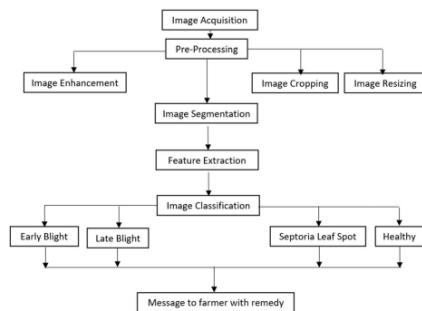
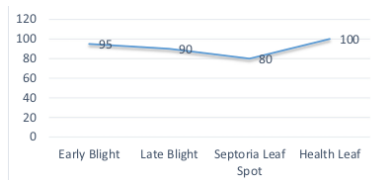
Student's Id and Name	Name: Mustain Murtaza Taib and ID: 18193103003 Name: Md Rakibul Islam and ID: 19201103097
Capstone Project Title	Tomato Leaf Disease Classification via Compact Convolutional Neural Networks with Transfer Learning and Feature Selection
Supervisor Name & Designation	Name: Mr.T.M. Amir - Ul - Haque Bhuiyan & Designation: Assistant Professor, Department of CSE, BUBT
Course Teacher's Name & Designation	Name: Khan Md. Hasib & Designation: Assistant Professor, Department of CSE, BUBT

Aspects	Paper # 1 (Title)																																	
Title / Question (What is problem statement?)	Tomato Leaf Disease Classification via Compact Convolutional Neural Networks with Transfer Learning and Feature Selection																																	
Objectives / Goal (What is looking for?)	Tomatoes are valuable vegetables and a major crop in many countries. his study proposes a pipeline using three compact convolutional neural networks (CNNs) and transfer learning to extract condensed features. The pipeline achieves high accuracy (99.92% and 99.90%). Six classifiers are utilized in the identification process,The experimental results demonstrate the competitive performance of the proposed pipeline compared to previous studies on tomato leaf disease classification.																																	
Methodology / Theory (How to find the solution?)	<ul style="list-style-type: none">Three compact CNNs (ResNet-18, ShuffleNet, and MobileNet) are then retrained using transfer learning, and deep features are extracted from each CNN. Six ML classifiers (Naïve Bayes, K-nearest neighbor, decision tree, linear discriminant analysis, support vector machine, and quadratic discriminant analysis) are used to classify tomato leaves into ten classes.																																	
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras,Tensorflow,pandas,numpy,matplotlib ,os.																																	
Test / Experiment How to test and characterize the design/prototype?	For the experimental work, the datasets were divided into the ratio of 80% and 20%. 80% of the datasets were used to train classification algorithms, and the remaining 20% used for testing purposes.classifiers.																																	
Simulation/Test Data (What parameters are determined?)	Datasets : Bacterial Spot,Early Blight,Healthy,Late Blight, Leaf Mold,Mosaic virus,Septoria Leaf Spot,Two Spotted Spider Mites,Target Spot,Yellow Leaf Curl Virus.																																	
Result / Conclusion (What was the final result?)	<p>The RF classifier observed a maximum accuracy of 97.62%.</p> <table><thead><tr><th>Model</th><th>Features</th><th>Accuracy</th></tr></thead><tbody><tr><td>ResNet-50</td><td>Features of ResNet-50</td><td>97.0%</td></tr><tr><td>U-Net</td><td>ResNet-50</td><td>97.11%</td></tr><tr><td>Customized CNN</td><td>Customized CNN</td><td>99.3%</td></tr><tr><td>Customized CNN</td><td>Customized CNN</td><td>98.70%</td></tr><tr><td>Fine-tuned MobileNet</td><td>Features of MobileNet</td><td>90.3%</td></tr><tr><td>Spatial attention with CNN</td><td>Fully connected layer</td><td>95.20%</td></tr><tr><td>VGG16</td><td>Features of VGG16</td><td>96.19%</td></tr><tr><td>Multinomial Logistic regression</td><td>MobileNetV2 or NASNetMobile</td><td>97%</td></tr><tr><td>EfficientNet-B0</td><td>EfficientNet-B0</td><td>98.60%</td></tr><tr><td>KNN</td><td>Fully connected layer (MobileNet + ShuffleNet + ResNet-18) + hybrid FS</td><td>99.92%</td></tr></tbody></table>	Model	Features	Accuracy	ResNet-50	Features of ResNet-50	97.0%	U-Net	ResNet-50	97.11%	Customized CNN	Customized CNN	99.3%	Customized CNN	Customized CNN	98.70%	Fine-tuned MobileNet	Features of MobileNet	90.3%	Spatial attention with CNN	Fully connected layer	95.20%	VGG16	Features of VGG16	96.19%	Multinomial Logistic regression	MobileNetV2 or NASNetMobile	97%	EfficientNet-B0	EfficientNet-B0	98.60%	KNN	Fully connected layer (MobileNet + ShuffleNet + ResNet-18) + hybrid FS	99.92%
Model	Features	Accuracy																																
ResNet-50	Features of ResNet-50	97.0%																																
U-Net	ResNet-50	97.11%																																
Customized CNN	Customized CNN	99.3%																																
Customized CNN	Customized CNN	98.70%																																
Fine-tuned MobileNet	Features of MobileNet	90.3%																																
Spatial attention with CNN	Fully connected layer	95.20%																																
VGG16	Features of VGG16	96.19%																																
Multinomial Logistic regression	MobileNetV2 or NASNetMobile	97%																																
EfficientNet-B0	EfficientNet-B0	98.60%																																
KNN	Fully connected layer (MobileNet + ShuffleNet + ResNet-18) + hybrid FS	99.92%																																
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	In the future, the work could be extended for different applications like weather forecasting, military applications, food predictions, etc.																																	
Terminology (List the common basic words frequently used in this research field)	Artificial intelligence. Machine learning · Internet of things (IoT) · Healthcare · Fog computing · Learning classifier. KNN-K-Nearest Neighbor RF-Random Forest, SVM- Support Vector Machine																																	

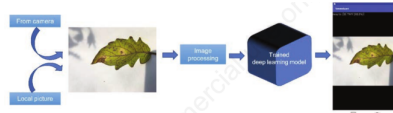
<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • Amin et al. proposed a system to classify people with heart disease and healthy people and had an accuracy of 89% • Samuel et al. proposed a system to predict heart failure risks, decision support systems based on artificial neural networks and had an accuracy of 91.10%
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>Along with the seven machine learning classification algorithms, I would use (analytic hierarchy process) AHP technique due to its simplicity, scalability, mathematical background, and ability to assess qualitative and quantitative factors to evaluate the effectiveness and efficacy of monitoring patients.</p>

Aspects	Paper # 2 (Title)																																																															
Title / Question (What is problem statement?)	Design of Efficient Methods for the Detection of Tomato Leaf Disease Utilizing Proposed Ensemble CNN Model																																																															
Objectives / Goal (What is looking for?)	They introduce two novel convolutional neural network (CNN) models alongside four well-known CNN models, and employ fine-tuning and hyperparameter optimization techniques using particle swarm optimization (PSO) and grid search. Triple and quintuple ensemble models are created, and the dataset is classified using five-fold cross-validation. The experimental results demonstrate that the proposed ensemble models exhibit fast training and testing times, achieving exceptional classification performance with an accuracy of 99.60%. This research simplifies and expedites the early detection of plant diseases, aiding experts in preventing the spread of infections.																																																															
Methodology / Theory (How to find the solution?)	<ul style="list-style-type: none">This section covers methods for tomato plant leaf disease classification, utilizing CNNs for automatic feature extraction. EfficientNet optimizes network performance through composite scaling, while InceptionV3 and GoogleNet enhance architecture efficiency with parallel computing and reduced parameters.																																																															
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras,Tensorflow,pandas,numpy,matplotlib ,os.																																																															
Test / Experiment How to test and characterize the design/prototype?																																																																
Simulation/Test Data (What parameters are determined?)	Datasets : Bacterial Spot,Early Blight,Healthy,Late Blight, Leaf Mold,Mosaic virus,Septoria Leaf Spot,Two Spotted Spider Mites,Target Spot,Yellow Leaf Curl Virus.																																																															
Result / Conclusion (What was the final result?)	<table><tr><th>Deep Learning Architecturs</th><th>Parameters</th><th>Storage Space</th><th>Epochs Required to Train the Model</th><th>Training Time (in Hours)</th><th>Testing Time (in Seconds)</th><th>Training Accuracy</th><th>Validation Accuracy</th><th>Testing Accuracy</th></tr><tr><td>Mobile NetV3small</td><td>1,234,547</td><td>55.6 MB</td><td>15</td><td>2.76</td><td>5.70</td><td>98.91%</td><td>99.9%</td><td>98.30%</td></tr><tr><td>EfficientNet V2L</td><td>117,911,203</td><td>1.4 GB</td><td>15</td><td>15.39</td><td>38.68</td><td>99.17%</td><td>99.71%</td><td>98.79%</td></tr><tr><td>InceptionV3</td><td>22,065,443</td><td>266 MB</td><td>15</td><td>3.51</td><td>10.35</td><td>97.06%</td><td>95.82%</td><td>96.1%</td></tr><tr><td>MobileNetV2</td><td>2,422,339</td><td>29.7 MB</td><td>15</td><td>2.32</td><td>7.0</td><td>98.34%</td><td>96.94%</td><td>96.99%</td></tr><tr><td>CNN1</td><td>5,108,426</td><td>5.1 MB</td><td>50</td><td>1.38</td><td>3.69</td><td>96.39%</td><td>99.03%</td><td>95.45%</td></tr><tr><td>CNN2</td><td>494,218</td><td>6 MB</td><td>50</td><td>1.37</td><td>3.49</td><td>97.61%</td><td>99.48%</td><td>96.87%</td></tr></table>	Deep Learning Architecturs	Parameters	Storage Space	Epochs Required to Train the Model	Training Time (in Hours)	Testing Time (in Seconds)	Training Accuracy	Validation Accuracy	Testing Accuracy	Mobile NetV3small	1,234,547	55.6 MB	15	2.76	5.70	98.91%	99.9%	98.30%	EfficientNet V2L	117,911,203	1.4 GB	15	15.39	38.68	99.17%	99.71%	98.79%	InceptionV3	22,065,443	266 MB	15	3.51	10.35	97.06%	95.82%	96.1%	MobileNetV2	2,422,339	29.7 MB	15	2.32	7.0	98.34%	96.94%	96.99%	CNN1	5,108,426	5.1 MB	50	1.38	3.69	96.39%	99.03%	95.45%	CNN2	494,218	6 MB	50	1.37	3.49	97.61%	99.48%	96.87%
Deep Learning Architecturs	Parameters	Storage Space	Epochs Required to Train the Model	Training Time (in Hours)	Testing Time (in Seconds)	Training Accuracy	Validation Accuracy	Testing Accuracy																																																								
Mobile NetV3small	1,234,547	55.6 MB	15	2.76	5.70	98.91%	99.9%	98.30%																																																								
EfficientNet V2L	117,911,203	1.4 GB	15	15.39	38.68	99.17%	99.71%	98.79%																																																								
InceptionV3	22,065,443	266 MB	15	3.51	10.35	97.06%	95.82%	96.1%																																																								
MobileNetV2	2,422,339	29.7 MB	15	2.32	7.0	98.34%	96.94%	96.99%																																																								
CNN1	5,108,426	5.1 MB	50	1.38	3.69	96.39%	99.03%	95.45%																																																								
CNN2	494,218	6 MB	50	1.37	3.49	97.61%	99.48%	96.87%																																																								
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	They didn't face any challenges																																																															
Terminology (List the common basic words frequently used in this research field)	CNN; deep learning; fine tuning; hyperparameter optimization; tomato disease;																																																															

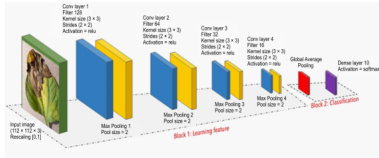
<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<p>It gets highest accuray 99.9% using MobileNetV3Small which better than this paper ”Yu, Y.; Samali, B.; Rashidi, M.; Mohammadi, M.; Nguyen, T.N.; Zhang, G. Vision-based concrete crack detection using a hybrid framework considering noise effect. J. Build. Eng. 2022, 61, 105246. [CrossRef]”</p>
<p>Review Outcome</p>	<p>This paper was great validation accuracy using mobilenetv3 and they used other model like Ineptionv3 and MobileNetv2</p>

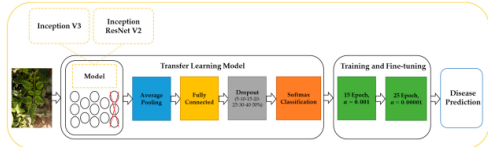
Aspects	Paper # 3 (Title)										
Title / Question (What is problem statement?)	Image processing based system for the detection, identification and treatment of tomato leaf diseases.										
Objectives / Goal (What is looking for?)	Early detection and treatment of leaf diseases in tomato plants are crucial for optimal production. Traditional manual methods are insufficient, leading to losses in quality and quantity. This paper presents an image processing-based technique using statistical features and SVM for automatic disease detection and treatment. Experimental results show high accuracy, and the method is implemented as a cell phone application.										
Methodology / Theory (How to find the solution?)	This paper proposes an image processing-based technique for automatic disease detection and treatment in tomato plants using statistical features, segmentation, and SVM. Experimental results show high accuracy, and the approach is implemented as a cell phone application, collecting data from tomato fields. Pre-processing includes cropping, resizing, and enhancement. The images are pre-processed, labeled, and stored, with SVM and K-nearest neighbor algorithm used for supervised learning, and GLCM employed for disease classification.										
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras, Tensorflow, pandas, numpy, matplotlib, os.										
Test / Experiment How to test and characterize the design/prototype?	 <pre> graph TD A[Image Acquisition] --> B[Pre-Processing] B --> C[Image Enhancement] B --> D[Image Cropping] B --> E[Image Resizing] C --> F[Image Segmentation] D --> F E --> F F --> G[Feature Extraction] G --> H[Image Classification] H --> I[Early Blight] H --> J[Late Blight] H --> K[Septoria Leaf Spot] H --> L[Healthy] I --> M[Message to farmer with remedy] J --> M K --> M L --> M </pre>										
Simulation/Test Data (What parameters are determined?)	Datasets : Bacterial Spot, Early Blight, Healthy, Late Blight, Leaf Mold, Mosaic virus, Septoria Leaf Spot, Two Spotted Spider Mites, Target Spot, Yellow Leaf Curl Virus.										
Result / Conclusion (What was the final result?)	 <table border="1"> <thead> <tr> <th>Disease Category</th> <th>Accuracy (%)</th> </tr> </thead> <tbody> <tr> <td>Early Blight</td> <td>95</td> </tr> <tr> <td>Late Blight</td> <td>90</td> </tr> <tr> <td>Septoria Leaf Spot</td> <td>80</td> </tr> <tr> <td>Healthy Leaf</td> <td>100</td> </tr> </tbody> </table>	Disease Category	Accuracy (%)	Early Blight	95	Late Blight	90	Septoria Leaf Spot	80	Healthy Leaf	100
Disease Category	Accuracy (%)										
Early Blight	95										
Late Blight	90										
Septoria Leaf Spot	80										
Healthy Leaf	100										
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	They didn't face any challenges										
Terminology (List the common basic words frequently used in this research field)	Image processing . Tomato crop . Leaf diseases . Early blight, late blight . Septoria leaf spot										

Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)	used (SVMs) with alternate kernel functions and get 99.5%
Review Outcome	This paper didn't use updated model

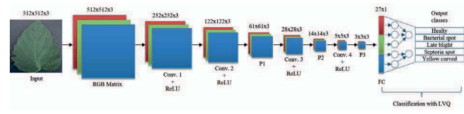
Aspects	Paper # 4 (Title)																												
Title / Question (What is problem statement?)	Tomato leaf diseases recognition based on deep convolutional neural networks																												
Objectives / Goal (What is looking for?)	work proposes a deep learning-based model for tomato leaf disease identification, utilizing in-house and public image databases. VGG16, Inceptionv3, and Resnet50 architectures were trained and tested, resulting in TomatoGuard, an Android application with 99% test accuracy. TomatoGuard outperforms the widely used Plantix app for general-purpose plant disease detection, addressing the need for practical field applications.																												
Methodology / Theory (How to find the solution?)	Fungi (e.g., early blight, septoria leaf spot, target spot, leaf mould), bacterial spot, late blight (caused by mould), tomato yellow leaf curl virus, and mites cause specific tomato diseases. Image preprocessing enhances data through techniques like rotation, flipping, mirroring, brightness adjustment, and cropping. Augmentation increases dataset diversity, and the data is split into training and testing sets.																												
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras,Tensorflow,pandas,numpy,matplotlib ,os.																												
Test / Experiment How to test and characterize the design/prototype?																													
Simulation/Test Data (What parameters are determined?)	Datasets : Bacterial Spot,Early Blight,Healthy,Late Blight, Leaf Mold,Mosaic virus,Septoria Leaf Spot,Two Spotted Spider Mites,Target Spot,Yellow Leaf Curl Virus.																												
Result / Conclusion (What was the final result?)	<p>Table 3. Model convergence result.</p> <table><tr><th>Model</th><th>Min validation loss</th><th>Regular Epoch</th><th>Test accuracy %</th><th>Min validation loss</th><th>Add early stop Epoch</th><th>Test accuracy %</th></tr><tr><td>VGG16</td><td>0.00322</td><td>288</td><td>99.75</td><td>0.00356</td><td>162</td><td>99.62</td></tr><tr><td>Inception_v3</td><td>0.00491</td><td>273</td><td>99.42</td><td>0.00694</td><td>177</td><td>99.75</td></tr><tr><td>Resnet50</td><td>0.00281</td><td>246</td><td>98.92</td><td>0.00779</td><td>108</td><td>99.50</td></tr></table>	Model	Min validation loss	Regular Epoch	Test accuracy %	Min validation loss	Add early stop Epoch	Test accuracy %	VGG16	0.00322	288	99.75	0.00356	162	99.62	Inception_v3	0.00491	273	99.42	0.00694	177	99.75	Resnet50	0.00281	246	98.92	0.00779	108	99.50
Model	Min validation loss	Regular Epoch	Test accuracy %	Min validation loss	Add early stop Epoch	Test accuracy %																							
VGG16	0.00322	288	99.75	0.00356	162	99.62																							
Inception_v3	0.00491	273	99.42	0.00694	177	99.75																							
Resnet50	0.00281	246	98.92	0.00779	108	99.50																							
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Early detection of tomato leaf diseases is crucial for reducing pesticide usage. Transfer learning, particularly fine-tuning, is a commonly used technique where models trained on one task are repurposed for related tasks. Models like VGG16, Inceptionv3, and ResNet50 have shown impressive performance in computer vision tasks such as ImageNet.																												
Terminology (List the common basic words frequently used in this research field)	image augmented; deep learning; image classification; android application.																												

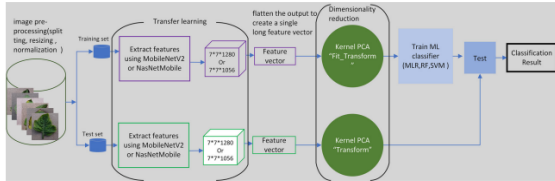
Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)	used VGG16 and get test accuracy 99.62%
Review Outcome	This paper didn't use updated model

Aspects	Paper # 5 (Title)
Title / Question (What is problem statement?)	Monitoring Tomato Leaf Disease through Convolutional Neural Networks
Objectives / Goal (What is looking for?)	Mexico's agricultural sector, particularly tomatoes, contributes significantly to the economy. Disease identification in crops, addressed using deep learning techniques, is crucial for improving yields. This study proposes a high-performing convolutional neural network model that classifies tomato leaf diseases with over 99% accuracy.
Methodology / Theory (How to find the solution?)	Proposed architecture for tomato leaf disease detection: Input: tomato leaf images. Output: disease labels, predicted values, and prediction percentages. Steps: dataset creation, architecture design, dataset distribution, model training and evaluation. Dataset: 13,500 images of 10 disease categories and healthy leaves. Overfitting prevention: GAN generates synthetic samples. CNN architecture: 4 convolutional layers, MaxPooling layers. Evaluation: k-fold cross-validation. Training: Adam optimizer, categorical crossentropy loss function.
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras, Tensorflow, pandas, numpy, matplotlib, os.
Test / Experiment How to test and characterize the design/prototype?	
Simulation/Test Data (What parameters are determined?)	Datasets : Bacterial Spot, Early Blight, Healthy, Late Blight, Leaf Mold, Mosaic virus, Septoria Leaf Spot, Two Spotted Spider Mites, Target Spot, Yellow Leaf Curl Virus.
Result / Conclusion (What was the final result?)	ResNet - 97.6 , VGG16Net - 98.77, Inception-v3-Net - 98.8, AlexNet - 99.64
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	New innovative techniques are needed to address the challenges and trends in agricultural production. This requires higher accuracy levels and improved detection methods, considering the new vision of agricultural monitoring.
Terminology (List the common basic words frequently used in this research field)	convolutional neural networks; deep learning; disease classification; generative adversarial network; tomato leaf.
Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)	AlexNet - 99.64%
Review Outcome	This paper didn't use updated model

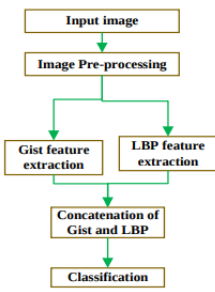
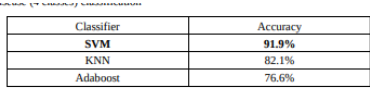
Aspects	Paper # 6 (Title)																																																																																																																																											
Title / Question (What is problem statement?)	Smart Detection of Tomato Leaf Diseases Using Transfer Learning-Based Convolutional Neural Networks																																																																																																																																											
Objectives / Goal (What is looking for?)	Data augmentation techniques, including horizontal flipping, rotation, and zooming, were applied to tomato disease leaf images. The augmented images were used for further analysis. A convolutional neural network (CNN) architecture was employed, consisting of convolutional, batch normalization, activation, pooling, and fully connected layers. Pre-trained Inception V3 and Inception ResNet V2 models were used, with transfer learning applied to improve model performance. Dropout layers were utilized to address background noise. The CNN models were trained using cross-entropy loss and the Adam optimizer. This approach outperformed traditional feature extraction methods for plant disease identification.																																																																																																																																											
Methodology / Theory (How to find the solution?)	Proposed architecture for tomato leaf disease detection: Input: tomato leaf images. Output: disease labels, predicted values, and prediction percentages. Steps: dataset creation, architecture design, dataset distribution, model training and evaluation. Dataset: 13,500 images of 10 disease categories and healthy leaves. Overfitting prevention: GAN generates synthetic samples. CNN architecture: 4 convolutional layers, MaxPooling layers. Evaluation: k-fold cross-validation. Training: Adam optimizer, categorical crossentropy loss function.																																																																																																																																											
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras,Tensorflow,pandas,numpy,matplot ,os.																																																																																																																																											
Test / Experiment How to test and characterize the design/prototype?	<div></div> <p>Figure 4. CNN architecture to develop a pre-trained model for tomato disease diagnosis.</p>																																																																																																																																											
Simulation/Test Data (What parameters are determined?)	Datasets : Bacterial Spot,Early Blight,Healthy,Late Blight, Leaf Mold,Mosaic virus,Septoria Leaf Spot,Two Spotted Spider Mites,Target Spot,Yellow Leaf Curl Virus.																																																																																																																																											
Result / Conclusion (What was the final result?)	<table><tr><th rowspan="2">Drop Out (%)</th><th colspan="2">Train</th><th colspan="2">Validation</th><th colspan="2">Test</th></tr><tr><th>Accuracy (%)</th><th>Loss</th><th>Accuracy (%)</th><th>Loss</th><th>Accuracy (%)</th><th>Loss</th></tr><tr><td colspan="7">Inception V3 model</td></tr><tr><td>5</td><td>99.33</td><td>0.0166</td><td>97.56</td><td>0.0624</td><td>97.85</td><td>0.0685</td></tr><tr><td>10</td><td>99.57</td><td>0.0103</td><td>98.12</td><td>0.0466</td><td>98.83</td><td>0.0460</td></tr><tr><td>15</td><td>99.95</td><td>0.0034</td><td>99.06</td><td>0.0311</td><td>98.83</td><td>0.0342</td></tr><tr><td>20</td><td>99.55</td><td>0.0107</td><td>97.75</td><td>0.0446</td><td>99.02</td><td>0.0366</td></tr><tr><td>25</td><td>99.90</td><td>0.0046</td><td>98.87</td><td>0.0321</td><td>98.63</td><td>0.0411</td></tr><tr><td>30</td><td>99.81</td><td>0.0059</td><td>98.69</td><td>0.0361</td><td>98.63</td><td>0.0234</td></tr><tr><td>40</td><td>99.83</td><td>0.0061</td><td>98.12</td><td>0.0332</td><td>98.63</td><td>0.0388</td></tr><tr><td>50</td><td>99.78</td><td>0.0063</td><td>98.69</td><td>0.0252</td><td>99.22</td><td>0.0318</td></tr><tr><td colspan="7">Inception ResNet V2 model</td></tr><tr><td>5</td><td>99.83</td><td>0.0046</td><td>98.31</td><td>0.0598</td><td>99.02</td><td>0.0438</td></tr><tr><td>10</td><td>99.88</td><td>0.0037</td><td>98.87</td><td>0.0314</td><td>98.83</td><td>0.0322</td></tr><tr><td>15</td><td>99.93</td><td>0.0022</td><td>98.87</td><td>0.0277</td><td>99.22</td><td>0.0399</td></tr><tr><td>20</td><td>99.93</td><td>0.0021</td><td>98.69</td><td>0.0392</td><td>98.83</td><td>0.0396</td></tr><tr><td>25</td><td>99.78</td><td>0.0065</td><td>98.69</td><td>0.0457</td><td>99.22</td><td>0.0522</td></tr><tr><td>30</td><td>99.95</td><td>0.0024</td><td>99.06</td><td>0.0398</td><td>99.02</td><td>0.0495</td></tr><tr><td>40</td><td>99.40</td><td>0.0128</td><td>98.50</td><td>0.0632</td><td>98.83</td><td>0.0636</td></tr><tr><td>50</td><td>99.83</td><td>0.0058</td><td>98.50</td><td>0.0379</td><td>99.02</td><td>0.0467</td></tr></table>	Drop Out (%)	Train		Validation		Test		Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)	Loss	Inception V3 model							5	99.33	0.0166	97.56	0.0624	97.85	0.0685	10	99.57	0.0103	98.12	0.0466	98.83	0.0460	15	99.95	0.0034	99.06	0.0311	98.83	0.0342	20	99.55	0.0107	97.75	0.0446	99.02	0.0366	25	99.90	0.0046	98.87	0.0321	98.63	0.0411	30	99.81	0.0059	98.69	0.0361	98.63	0.0234	40	99.83	0.0061	98.12	0.0332	98.63	0.0388	50	99.78	0.0063	98.69	0.0252	99.22	0.0318	Inception ResNet V2 model							5	99.83	0.0046	98.31	0.0598	99.02	0.0438	10	99.88	0.0037	98.87	0.0314	98.83	0.0322	15	99.93	0.0022	98.87	0.0277	99.22	0.0399	20	99.93	0.0021	98.69	0.0392	98.83	0.0396	25	99.78	0.0065	98.69	0.0457	99.22	0.0522	30	99.95	0.0024	99.06	0.0398	99.02	0.0495	40	99.40	0.0128	98.50	0.0632	98.83	0.0636	50	99.83	0.0058	98.50	0.0379	99.02	0.0467
Drop Out (%)	Train		Validation		Test																																																																																																																																							
	Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)	Loss																																																																																																																																						
Inception V3 model																																																																																																																																												
5	99.33	0.0166	97.56	0.0624	97.85	0.0685																																																																																																																																						
10	99.57	0.0103	98.12	0.0466	98.83	0.0460																																																																																																																																						
15	99.95	0.0034	99.06	0.0311	98.83	0.0342																																																																																																																																						
20	99.55	0.0107	97.75	0.0446	99.02	0.0366																																																																																																																																						
25	99.90	0.0046	98.87	0.0321	98.63	0.0411																																																																																																																																						
30	99.81	0.0059	98.69	0.0361	98.63	0.0234																																																																																																																																						
40	99.83	0.0061	98.12	0.0332	98.63	0.0388																																																																																																																																						
50	99.78	0.0063	98.69	0.0252	99.22	0.0318																																																																																																																																						
Inception ResNet V2 model																																																																																																																																												
5	99.83	0.0046	98.31	0.0598	99.02	0.0438																																																																																																																																						
10	99.88	0.0037	98.87	0.0314	98.83	0.0322																																																																																																																																						
15	99.93	0.0022	98.87	0.0277	99.22	0.0399																																																																																																																																						
20	99.93	0.0021	98.69	0.0392	98.83	0.0396																																																																																																																																						
25	99.78	0.0065	98.69	0.0457	99.22	0.0522																																																																																																																																						
30	99.95	0.0024	99.06	0.0398	99.02	0.0495																																																																																																																																						
40	99.40	0.0128	98.50	0.0632	98.83	0.0636																																																																																																																																						
50	99.83	0.0058	98.50	0.0379	99.02	0.0467																																																																																																																																						
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Team didnt find any challenges xi																																																																																																																																											
Terminology (List the common	deep learning; convolutional neural networks; inception V3;																																																																																																																																											

Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)	Validation accuracy is 98.69% Test Accuracy is 99.22% Train Accuracy is 99.78% In Inceptionv3
Review Outcome	This paper didn't use updated model

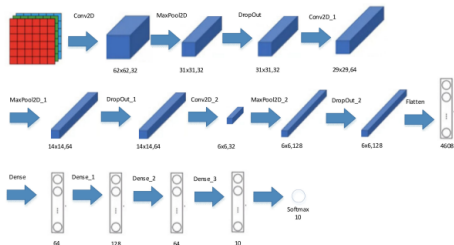
Aspects	Paper # 7 (Title)																																																	
Title / Question (What is problem statement?)	Plant Leaf Disease Detection and Classification based on CNN with LVQ Algorithm																																																	
Objectives / Goal (What is looking for?)	This paper proposes a method for tomato leaf disease detection and classification using a Convolutional Neural Network (CNN) and Learning Vector Quantization (LVQ) algorithm. The dataset consists of 500 images with four disease symptoms. The CNN model extracts features from RGB channels, and the LVQ is used for network training. The experimental results demonstrate accurate recognition of the four tomato leaf diseases. Key terms: Leaf Disease Detection, Leaf Disease Classification, CNN, LVQ.																																																	
Methodology / Theory (How to find the solution?)	CNN extracts features from an input image using convolution layer.The pooling layer follows the convolution layer and reduces the size of the output matrix. A commonly used 2x2 filter is applied, with options like max pooling, average pooling, or L2-norm pooling.In this study, the LVQ algorithm is used for data classification.																																																	
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras, Tensorflow, pandas, numpy, matplotlib ,os.																																																	
Test / Experiment How to test and characterize the design/prototype?	<div></div> <p>Fig. 6. Architecture of the proposed method.</p>																																																	
Simulation/Test Data (What parameters are determined?)	Datasets : Healthy, Bacterial spot, Late blight, Septoria spot, Yellow curved																																																	
Result / Conclusion (What was the final result?)	<div>TABLE I CLASSIFICATION RESULTS AS CONFUSION MATRIX.</div> <table><tr><th>Leaf Disease</th><th>Healthy</th><th>Bacterial spot</th><th>Late blight</th><th>Septoria spot</th><th>Yellow curved</th><th>Accuracy</th></tr><tr><td>Healthy</td><td>18</td><td>0</td><td>0</td><td>0</td><td>2</td><td>90%</td></tr><tr><td>Bacterial spot</td><td>0</td><td>18</td><td>0</td><td>0</td><td>2</td><td>90%</td></tr><tr><td>Late blight</td><td>0</td><td>0</td><td>17</td><td>0</td><td>3</td><td>85%</td></tr><tr><td>Septoria spot</td><td>1</td><td>0</td><td>0</td><td>16</td><td>3</td><td>80%</td></tr><tr><td>Yellow curved</td><td>0</td><td>0</td><td>0</td><td>3</td><td>17</td><td>85%</td></tr><tr><td>Average</td><td></td><td></td><td></td><td></td><td></td><td>86%</td></tr></table>	Leaf Disease	Healthy	Bacterial spot	Late blight	Septoria spot	Yellow curved	Accuracy	Healthy	18	0	0	0	2	90%	Bacterial spot	0	18	0	0	2	90%	Late blight	0	0	17	0	3	85%	Septoria spot	1	0	0	16	3	80%	Yellow curved	0	0	0	3	17	85%	Average						86%
Leaf Disease	Healthy	Bacterial spot	Late blight	Septoria spot	Yellow curved	Accuracy																																												
Healthy	18	0	0	0	2	90%																																												
Bacterial spot	0	18	0	0	2	90%																																												
Late blight	0	0	17	0	3	85%																																												
Septoria spot	1	0	0	16	3	80%																																												
Yellow curved	0	0	0	3	17	85%																																												
Average						86%																																												
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Team didnt find any challenges																																																	
Terminology (List the common basic words frequently used in this research field)	Leaf Disease Detection, Leaf Disease Classification, Convolutional Neural Network (CNN), Learning Vector Quantization (LVQ)																																																	
Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)	Healthy class got 90% In Inceptionv3																																																	
Review Outcome	This paper didn't use updated model																																																	

Aspects	Paper # 8 (Title)																																																																
Title / Question (What is problem statement?)	Tomato leaf disease classification by exploiting transfer learning and feature concatenation																																																																
Objectives / Goal (What is looking for?)	The authors propose a method for automating tomato leaf disease classification using transfer learning and feature concatenation. Pre-trained kernels from MobileNetV2 and NASNetMobile are used to extract features, which are then reduced in dimensionality using kernel principal component analysis. The concatenated features are fed into a conventional learning algorithm, with multinomial logistic regression achieving an average accuracy of 97%.																																																																
Methodology / Theory (How to find the solution?)	They selected 1152 tomato leaf images, divided into one healthy class and five unhealthy classes, to extract features. The images were resized to 224x224 pixels and normalized between 0 and 1. Transfer learning is used to adapt pre-trained models for image classification, specifically MobileNetV2 and NASNetMobile, as feature extractors without their final classification layer.																																																																
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras, Tensorflow, pandas, numpy, matplotlib ,os.																																																																
Test / Experiment How to test and characterize the design/prototype?																																																																	
Simulation/Test Data (What parameters are determined?)	Datasets : Healthy, Bacterial spot, Late blight, Septoria spot, Yellow curved																																																																
Result / Conclusion (What was the final result?)	<table><tr><th>Related work</th><th>Features</th><th>Model</th><th>Dataset</th><th>Number of classes</th><th>Number of training data</th><th>Original classification accuracy (%)</th><th>Classification accuracy on our dataset with six classes (%)</th></tr><tr><td>Huang et al. [14]</td><td>SIFT Texture feature + statistical colour features</td><td>SVM</td><td>PlantVillage</td><td>7</td><td>3,535</td><td>85</td><td>—</td></tr><tr><td>Zaki et al. [20]</td><td>Fine-tune MobileNetV2</td><td>MobileNetV2</td><td>PlantVillage</td><td>4</td><td>3,471</td><td>95.6</td><td>90.3</td></tr><tr><td>Basaviah et al. [15]</td><td>Ha Moments, Local Binary Pattern features, and colour histograms.</td><td>Random forest and decision tree</td><td>PlantVillage</td><td>5</td><td>300</td><td>90 for decision tree and 94 for RF</td><td>—</td></tr><tr><td>Agarwal et al. [21]</td><td>Model-based</td><td>Typical CNN model</td><td>PlantVillage</td><td>10</td><td>10,000</td><td>91.2</td><td>87</td></tr><tr><td>Nithish et al. [25]</td><td>Model-based</td><td>ResNet-50</td><td>PlantVillage</td><td>6</td><td>12,206</td><td>97</td><td>81.4</td></tr><tr><td>Proposed method 1</td><td>Fixed feature extractor of MobileNetV2 or NASNetMobile</td><td>Multinomial logistic regression (MLR)</td><td>PlantVillage</td><td>6</td><td>1,152</td><td>93.8 for MobileNetV2 and 92.4 for NASNetMobile</td><td>93.8 for MobileNetV2 and 92.4 for NASNetMobile</td></tr><tr><td>Proposed method 2</td><td>Combination features of MobileNetV2 +</td><td>Multinomial logistic regression</td><td>PlantVillage</td><td>6</td><td>1,152</td><td>97</td><td>97</td></tr></table>	Related work	Features	Model	Dataset	Number of classes	Number of training data	Original classification accuracy (%)	Classification accuracy on our dataset with six classes (%)	Huang et al. [14]	SIFT Texture feature + statistical colour features	SVM	PlantVillage	7	3,535	85	—	Zaki et al. [20]	Fine-tune MobileNetV2	MobileNetV2	PlantVillage	4	3,471	95.6	90.3	Basaviah et al. [15]	Ha Moments, Local Binary Pattern features, and colour histograms.	Random forest and decision tree	PlantVillage	5	300	90 for decision tree and 94 for RF	—	Agarwal et al. [21]	Model-based	Typical CNN model	PlantVillage	10	10,000	91.2	87	Nithish et al. [25]	Model-based	ResNet-50	PlantVillage	6	12,206	97	81.4	Proposed method 1	Fixed feature extractor of MobileNetV2 or NASNetMobile	Multinomial logistic regression (MLR)	PlantVillage	6	1,152	93.8 for MobileNetV2 and 92.4 for NASNetMobile	93.8 for MobileNetV2 and 92.4 for NASNetMobile	Proposed method 2	Combination features of MobileNetV2 +	Multinomial logistic regression	PlantVillage	6	1,152	97	97
Related work	Features	Model	Dataset	Number of classes	Number of training data	Original classification accuracy (%)	Classification accuracy on our dataset with six classes (%)																																																										
Huang et al. [14]	SIFT Texture feature + statistical colour features	SVM	PlantVillage	7	3,535	85	—																																																										
Zaki et al. [20]	Fine-tune MobileNetV2	MobileNetV2	PlantVillage	4	3,471	95.6	90.3																																																										
Basaviah et al. [15]	Ha Moments, Local Binary Pattern features, and colour histograms.	Random forest and decision tree	PlantVillage	5	300	90 for decision tree and 94 for RF	—																																																										
Agarwal et al. [21]	Model-based	Typical CNN model	PlantVillage	10	10,000	91.2	87																																																										
Nithish et al. [25]	Model-based	ResNet-50	PlantVillage	6	12,206	97	81.4																																																										
Proposed method 1	Fixed feature extractor of MobileNetV2 or NASNetMobile	Multinomial logistic regression (MLR)	PlantVillage	6	1,152	93.8 for MobileNetV2 and 92.4 for NASNetMobile	93.8 for MobileNetV2 and 92.4 for NASNetMobile																																																										
Proposed method 2	Combination features of MobileNetV2 +	Multinomial logistic regression	PlantVillage	6	1,152	97	97																																																										
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Team didnt find any challenges																																																																
Terminology (List the common basic words frequently used in this research field)	Leaf Disease Detection, Leaf Disease Classification, Convolutional Neural Network (CNN), NASNetMobile , MobileNetV2																																																																

Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)	ResNet-50 was trained on a dataset containing 12206 images; however, when we trained and tested it on our dataset (which contained only 1152 images), the accuracy declined dramatically from 97% to 81.4%.
Review Outcome	This paper didn't use updated model

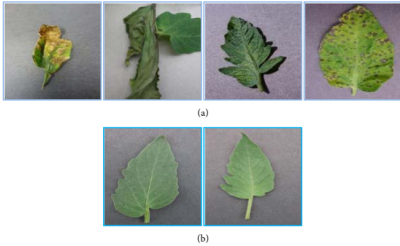
Aspects	Paper # 9 (Title)								
Title / Question (What is problem statement?)	Classification of Leaf Disease Using Global and Local Features								
Objectives / Goal (What is looking for?)	Leaf disease in plants can lead to crop productivity loss, highlighting the importance of timely diagnosis and proper care. This paper presents a computer vision-based method using Gist and LBP features for early disease detection, offering a time-efficient alternative to deep learning approaches. Gist provides a global image description, LBP is resilient to illumination changes and occlusions. Pre-processing and combining Gist and LBP features yield promising results, with SVM outperforming other machine learning algorithms in classifying plant leaf datasets.								
Methodology / Theory (How to find the solution?)	Gist Descriptors: Summarize gradient information using Gabor filters at multiple scales, locations, and orientations, resulting in a 128-value Gist feature vector. LBP (Local Binary Patterns): Robust to lighting variations, LBP encodes local structure by comparing 3x3 neighborhood values, generating binary patterns for labeling. SVM (Support Vector Machine): Versatile model for classification and regression, drawing a line or hyperplane to separate classes and maximize the margin between support vectors. KNN (K-Nearest Neighbors): Classifies data points based on proximity to K nearest neighbors. Adaboost: Ensemble method that iteratively trains weak learners on weighted data, transforming them into stronger models.								
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras, Tensorflow, pandas, numpy, matplotlib, os.								
Test / Experiment How to test and characterize the design/prototype?	 <pre> graph TD A[Input image] --> B[Image Pre-processing] B --> C[Gist feature extraction] B --> D[LBP feature extraction] C --> E[Concatenation of Gist and LBP] D --> E E --> F[Classification] </pre>								
Simulation/Test Data (What parameters are determined?)	Datasets : Healthy, Bacterial spot, Late blight, Septoria spot, Yellow curved								
Result / Conclusion (What was the final result?)	 <table border="1"> <thead> <tr> <th>Classifier</th><th>Accuracy</th></tr> </thead> <tbody> <tr> <td>SVM</td><td>91.9%</td></tr> <tr> <td>KNN</td><td>82.1%</td></tr> <tr> <td>Adaboost</td><td>76.6%</td></tr> </tbody> </table>	Classifier	Accuracy	SVM	91.9%	KNN	82.1%	Adaboost	76.6%
Classifier	Accuracy								
SVM	91.9%								
KNN	82.1%								
Adaboost	76.6%								
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Team didnt find any challenges								
Terminology (List the common basic words frequently used in this	Leaf disease, Gist, local binary pattern, machine learning.								

Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)	SVM 91.9% KNN 82.1% Adaboost 76.6%.
Review Outcome	This paper didn't use updated model

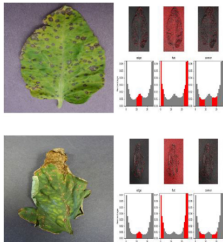
Aspects	Paper # 10 (Title)																																																									
Title / Question (What is problem statement?)	AlexNet Convolutional Neural Network for Disease Detection and Classification of Tomato Leaf																																																									
Objectives / Goal (What is looking for?)	With limited resources and increasing plant diseases, automation becomes crucial. This study employs an AlexNet-based CNN on Android to predict tomato diseases from leaf images. A dataset of 18,345 training and 4,585 testing samples, with ten disease labels and 64x64 RGB pixels, is used. The optimized model achieves high accuracy (98%), precision (0.98), recall (0.99), F1-score (0.98), and low loss (0.1331), ensuring precise and reliable disease classification.																																																									
Methodology / Theory (How to find the solution?)	Convolution layers use trainable kernels for convolutions and activation functions for feature map generation. Subsampling layers employ non-trainable kernels for downsampling using methods like average or max pooling, and can have trainable parameters. CNN learning involves selecting an architecture like modified AlexNet, using Adam optimizer with learning rate 0.005, and updating neuron weights with gradient descent. The input image size is 64x64 RGB pixels, employing ReLU activation in convolution with Alexnet architecture and max pooling layers, followed by flattening, dense layers with ReLU, and a Softmax activation for the output layer.																																																									
Software Tools (What program/software is used for design, coding and simulation?)	Google colab, keras,Tensorflow,pandas,numpy,matplotlib ,os.																																																									
Test / Experiment How to test and characterize the design/prototype?																																																										
Simulation/Test Data (What parameters are determined?)	Datasets : Healthy, Bacterial spot, Late blight, Septoria spot, Yellow curved																																																									
Result / Conclusion (What was the final result?)	<table><thead><tr><th>No</th><th>Disease</th><th>Precision</th><th>Recall</th><th>F-Measure</th><th>Accuracy</th></tr></thead><tbody><tr><td>1</td><td>Bacterial Spot</td><td>0.98</td><td>0.96</td><td>0.97</td><td rowspan="10">0.96</td></tr><tr><td>2</td><td>Early blight</td><td>0.94</td><td>0.94</td><td>0.94</td></tr><tr><td>3</td><td>Late blight</td><td>0.95</td><td>0.95</td><td>0.95</td></tr><tr><td>4</td><td>Leaves mold</td><td>0.98</td><td>0.97</td><td>0.97</td></tr><tr><td>5</td><td>Septoria leaf mold</td><td>0.95</td><td>0.95</td><td>0.95</td></tr><tr><td>6</td><td>Spider mites</td><td>0.95</td><td>0.96</td><td>0.96</td></tr><tr><td>7</td><td>Target spot</td><td>0.93</td><td>0.94</td><td>0.94</td></tr><tr><td>8</td><td>Yellow leaf curl virus</td><td>0.97</td><td>0.99</td><td>0.98</td></tr><tr><td>9</td><td>Tomato mosaic virus</td><td>0.98</td><td>1</td><td>0.99</td></tr><tr><td>10</td><td>Healthy</td><td>1</td><td>0.97</td><td>0.98</td></tr></tbody></table>	No	Disease	Precision	Recall	F-Measure	Accuracy	1	Bacterial Spot	0.98	0.96	0.97	0.96	2	Early blight	0.94	0.94	0.94	3	Late blight	0.95	0.95	0.95	4	Leaves mold	0.98	0.97	0.97	5	Septoria leaf mold	0.95	0.95	0.95	6	Spider mites	0.95	0.96	0.96	7	Target spot	0.93	0.94	0.94	8	Yellow leaf curl virus	0.97	0.99	0.98	9	Tomato mosaic virus	0.98	1	0.99	10	Healthy	1	0.97	0.98
No	Disease	Precision	Recall	F-Measure	Accuracy																																																					
1	Bacterial Spot	0.98	0.96	0.97	0.96																																																					
2	Early blight	0.94	0.94	0.94																																																						
3	Late blight	0.95	0.95	0.95																																																						
4	Leaves mold	0.98	0.97	0.97																																																						
5	Septoria leaf mold	0.95	0.95	0.95																																																						
6	Spider mites	0.95	0.96	0.96																																																						
7	Target spot	0.93	0.94	0.94																																																						
8	Yellow leaf curl virus	0.97	0.99	0.98																																																						
9	Tomato mosaic virus	0.98	1	0.99																																																						
10	Healthy	1	0.97	0.98																																																						
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Team didnt find any challenges																																																									
Terminology (List the common basic words frequently used in this research field)	AlexNet modification; tomato diseases; leaf image; AI.																																																									

xviii

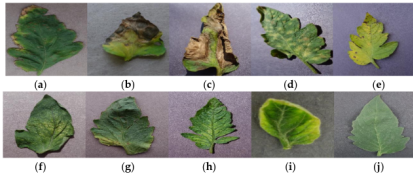
Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)	The research demonstrates that the CNN algorithm with a modified AlexNet architecture, combined with preprocessing and classification methods, achieves high accuracy (96%), precision (98%), recall (95%), and F-Measure (97
Review Outcome	This paper didn't use updated model

Aspects	Paper # 1 (Title)
Title / Question (What is problem statement?)	Deep Learning Convolution Neural Network to Detect and Classify Tomato Plant Leaf Diseases
Objectives / Goal (What is looking for?)	The goal is to develop an intelligent CNN-based technique for early detection and accurate classification of tomato plant diseases, with the aim of improving crop production and quality by providing farmers a reliable tool for disease detection and classification.
Methodology / Theory (How to find the solution?)	<p>The work was divided into three phases.</p> <ul style="list-style-type: none"> • Collect dataset from Plant Village • Design CNN architecture and use CNN for disease detection and classification • Extract relevant features and Train the CNN: • Evaluate classification accuracy and Validate with real images: Apply the CNN technique to real images obtained from a farm using a 5-megapixel camera.
Software Tools (What program/software is used for design, coding and simulation?)	Tensorflow,Python,Anaconda Notebook
Test / Experiment How to test and characterize the design/prototype?	The proposed methodology involves obtaining the plant village dataset, resizing the images, splitting them into training and testing sets, and classifying the images using a deep learning convolutional neural network (DLCNN) architecture. The experimental results show an accuracy of 96.43% for the proposed network.
Simulation/Test Data (What parameters are determined?)	<p>The simulation/test data consists of 6202 images from the plant village dataset, divided into six classes of tomato plant leaf diseases. The accuracy of the proposed network is evaluated using a training set of 4342 images and a testing set of 1860 images.</p>  <p>Figure 1. (a) Types tomato plant leaf diseases; (b) Healthy tomato plant leaf.</p>
Result / Conclusion (What was the final result?)	The proposed deep learning convolution neural network achieved a high classification accuracy of 96.34% for detecting and classifying tomato plant diseases, with a training accuracy of 99.36%.
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	In the future, the work could be extended for different applications like weather forecasting, military applications, food predictions, etc.
Terminology (List the common basic words frequently used in this research field)	onvolution Neural Network (CNN), Tomato Plant Leaf Diseases, Machine Learning, Early Detection

<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed articles aimed to classify plant diseases using deep learning models, with reported accuracies ranging from 89% to 96.77% on various datasets. • Rajasekaran Thangaraj's The Modified Xception model achieved an average detection accuracy of 99.55% in tomato leaf disease detection.
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>Based on the reviewed articles, the obtained knowledge will be used to inform the development of a new methodology for classifying plant diseases using a deep learning model, with a focus on dataset selection, network architecture, and performance evaluation.</p>

Aspects	Paper # 2 (Title)
Title / Question (What is problem statement?)	Tomato Leaf Disease Classification using Multiple Feature Extraction Techniques
Objectives / Goal (What is looking for?)	The objective is to develop a technique for identifying leaf disease in tomato plants by improving classification accuracy and reducing computational time.
Methodology / Theory (How to find the solution?)	<ul style="list-style-type: none"> • Collection of Database • The proposed method uses a fusion of multiple features (color histograms, Hu Moments, Haralick, and Local Binary Pattern) and employs random forest and decision tree classification algorithms for leaf disease identification in tomato plants.
Software Tools (What program/software is used for design, coding and simulation?)	Tensorflow,Python,Google Colab
Test / Experiment How to test and characterize the design/prototype?	The proposed method was evaluated by dividing the dataset into train and test folders, and then using decision tree and random forest classifiers to classify the test images. The classification accuracy was calculated using confusion matrices. The proposed method achieved an accuracy of 90% for the decision tree classifier and 94% for the random forest classifier. The results were compared with other state-of-the-art techniques, showing competitive performance.
Simulation/Test Data (What parameters are determined?)	<p>The dataset has five classes of tomato leaf diseases: Bacterial, Healthy, Septoria, Mosaic, and Yellow Curl. There are 60 images per class in the train folder (300 images in total) and 40 images per class in the test folder (200 images in total). The test images are classified using decision tree and random forest classifiers.</p>  <p>Fig 7. Leaf disease patterns and corresponding histograms plot</p>
Result / Conclusion (What was the final result?)	The proposed method achieved high classification accuracy (90% for decision tree classifier and 94% for random forest classifier) in detecting and classifying tomato leaf diseases, with the random forest classifier outperforming the decision tree classifier, while also offering reduced computational time.
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Future work includes expanding disease classification and improving accuracy using advanced features and developing an automated system for disease detection and action.
Terminology (List the common basic words frequently used in this research field)	Decision tree · Feature extraction · Leaf disease · Random forest


<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed articles gives 90% for decision tree classifier and 94% for random forest classifier • Thair A. Salih's articles aimed to classify plant diseases using deep learning models, with reported accuracies ranging from 89% to 96.77% on various datasets.
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>The obtained knowledge on disease detection and classification methods for tomato leaves will be used as a reference to develop an improved methodology for my own research project, incorporating advanced feature extraction techniques and focusing on expanding disease classification capabilities.</p>

Aspects	Paper # 3 (Title)
Title / Question (What is problem statement?)	Early Detection and Classification of Tomato Leaf Disease Using High-Performance Deep Neural Network
Objectives / Goal (What is looking for?)	The goal of the study is to develop a Convolutional Neural Network (CNN) model for accurately identifying and classifying tomato leaf diseases based on image analysis.
Methodology / Theory (How to find the solution?)	<ul style="list-style-type: none"> • Collection of dataset • Preprocessing of input images and segmentation of the targeted area.Extraction of features • Training the CNN model using a dataset consisting of 3000 images of tomato leaves affected by nine different diseases and healthy leaves.valuation of the model's performance and accuracy.
Software Tools (What program/software is used for design, coding and simulation?)	Tensorflow,Python,Anaconda Notebook
Test / Experiment How to test and characterize the design/prototype?	The design/prototype of the CNN model is tested and characterized by preparing a dataset of tomato leaf images, splitting it into training, validation, and testing sets. The model is then trained using the training set and evaluated on the validation set to assess its performance. Finally, the model is tested on unseen images from the testing set, analyzing its accuracy, speed, and robustness in identifying and classifying tomato leaf diseases. False positives, false negatives, and any limitations or challenges encountered during testing are also considered.
Simulation/Test Data (What parameters are determined?)	<p>The parameters determined during simulation/test data analysis for disease detection include accuracy, precision, recall, and F1-score, which measure the performance of the model in correctly identifying and classifying diseases based on test data.</p>  <p>Figure 1. Sample leaf image with disease and pathogen for (a) Bacterial_Spot(<i>Xanthomonas vesicatoria</i>), (b) Early_Blight(fungus <i>Alternaria solani</i>), (c) Late_Blight(<i>Phytophthora infestans</i>), (d) Leaf_Mold(<i>Cladosporium fulvum</i>), (e) Septoria_Leaf_Spot(fungus <i>Septoria lycopersici</i>), (f) Spider_Mites(<i>floridana</i>), (g) Target_Spot(fungus <i>Corynespora</i>), (h) Tomato_Mosaic_Virus(<i>Tobamovirus</i>), (i) Tomato_Yellow_Leaf_Curl_Virus(genus <i>Begomovirus</i>), (j) Healthy_Leaf.</p>
Result / Conclusion (What was the final result?)	The proposed model achieved an accuracy of 98.49% in accurately diagnosing and classifying tomato leaf diseases.
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	The challenges mentioned in the article include the need for a large amount of training data, the complexity of image pre-processing, and the inclusion of abiotic diseases in the model.
Terminology (List the common basic words frequently used in this research field)	image processing; convolution neural network; plant leaf disease; deep learning; artificial intelligence


<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed model achieved an accuracy of 98.49% in accurately diagnosing and classifying tomato leaf diseases. • Jagadeesh Basavaiah's articles gives 90% for decision tree classifier and 94% for random forest classifier
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>The obtained knowledge can be used to inform and guide the development of a new methodology for a research project focused on the detection and classification of plant diseases using deep learning models.</p>

Aspects	Paper # 4 (Title)																																																												
Title / Question (What is problem statement?)	Tomato Leaf Disease Diagnosis Based on Improved Convolution Neural Network by Attention Module																																																												
Objectives / Goal (What is looking for?)	The objective of this study is to propose a deep convolutional neural network with an attention mechanism for accurate and efficient diagnosis of tomato leaf diseases and provide a high-performance solution for crop diagnosis in real agricultural environments.																																																												
Methodology / Theory (How to find the solution?)	<ul style="list-style-type: none">• Build the Dataset and Data Augmentation• The proposed solution utilizes a deep convolutional neural network with attention mechanisms, incorporating residual blocks and attention extraction modules, to accurately extract complex features of various tomato leaf diseases.																																																												
Software Tools (What program/software is used for design, coding and simulation?)	Tensorflow,Python,Anaconda Notebook																																																												
Test / Experiment How to test and characterize the design/prototype?	To test and characterize the proposed deep convolutional neural network with attention mechanism for crop disease diagnosis, the model can be evaluated using a dataset of tomato leaf diseases and a dataset of grape leaf diseases. The model's performance can be measured by calculating the average identification accuracy, which should be compared with other existing models to assess its superiority in terms of network complexity and real-time performance.																																																												
Simulation/Test Data (What parameters are determined?)	<p>test data in this study include 4585 tomato leaf images with fixed size of 224x224 pixels, divided into 10 categories representing different diseases and health conditions of tomato plants.</p> <p>Table 1. Tomato leaf diseases dataset.</p> <table><tr><th>Type</th><th>Class</th><th>Origin Images</th><th>Augmentation Images</th><th>Trainset</th></tr><tr><td>bacterial spot</td><td>0</td><td>425</td><td>2125</td><td>1700</td></tr><tr><td>early blight</td><td>1</td><td>480</td><td>2400</td><td>1920</td></tr><tr><td>healthy</td><td>2</td><td>481</td><td>2405</td><td>1924</td></tr><tr><td>late blight</td><td>3</td><td>463</td><td>2315</td><td>1852</td></tr><tr><td>leaf mold</td><td>4</td><td>470</td><td>2350</td><td>1880</td></tr><tr><td>mosaic virus</td><td>5</td><td>448</td><td>2240</td><td>1792</td></tr><tr><td>septoria leaf spot</td><td>6</td><td>436</td><td>2180</td><td>1744</td></tr><tr><td>target spot</td><td>7</td><td>457</td><td>2285</td><td>1828</td></tr><tr><td>two-spotted spider mite</td><td>8</td><td>435</td><td>2175</td><td>1740</td></tr><tr><td>yellow leaf curl virus</td><td>9</td><td>490</td><td>2450</td><td>1960</td></tr><tr><td>Total</td><td></td><td>4585</td><td>22,925</td><td>18,340</td></tr></table>	Type	Class	Origin Images	Augmentation Images	Trainset	bacterial spot	0	425	2125	1700	early blight	1	480	2400	1920	healthy	2	481	2405	1924	late blight	3	463	2315	1852	leaf mold	4	470	2350	1880	mosaic virus	5	448	2240	1792	septoria leaf spot	6	436	2180	1744	target spot	7	457	2285	1828	two-spotted spider mite	8	435	2175	1740	yellow leaf curl virus	9	490	2450	1960	Total		4585	22,925	18,340
Type	Class	Origin Images	Augmentation Images	Trainset																																																									
bacterial spot	0	425	2125	1700																																																									
early blight	1	480	2400	1920																																																									
healthy	2	481	2405	1924																																																									
late blight	3	463	2315	1852																																																									
leaf mold	4	470	2350	1880																																																									
mosaic virus	5	448	2240	1792																																																									
septoria leaf spot	6	436	2180	1744																																																									
target spot	7	457	2285	1828																																																									
two-spotted spider mite	8	435	2175	1740																																																									
yellow leaf curl virus	9	490	2450	1960																																																									
Total		4585	22,925	18,340																																																									
Result / Conclusion (What was the final result?)	The SE-ResNet50 model achieved an average detection accuracy of 96.81% for the 10-class classification of tomato leaf diseases, outperforming other models and demonstrating its potential for early and automatic disease diagnosis in agriculture.																																																												
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Future work could focus on deploying the developed model to a greenhouse inspection robot and improving diagnostic performance by establishing a dataset in a real agricultural environment.																																																												
Terminology (List the common basic words frequently used in this research field)	tomato leaf disease; deep learning; convolutional neural network (cnn); attention mechanism; classification																																																												

<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed The SE-ResNet50 model achieved an average detection accuracy of 96.81% for the 10-class classification of tomato leaf diseases • Sali's articles aimed to classify plant diseases using deep learning models, with reported accuracies ranging from 89% to 96.77% on various datasets.
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>The obtained knowledge will be used to inform and reference the development of a separate and new methodology for my own research project in the field of crop disease detection and diagnosis.</p>

Aspects	Paper # 5 (Title)
Title / Question (What is problem statement?)	Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet
Objectives / Goal (What is looking for?)	The objective is to develop an effective method for tomato leaf disease recognition using a combination of ABCK-BWTR and B-ARNet.
Methodology / Theory (How to find the solution?)	<ul style="list-style-type: none"> • Collection of dataset • The proposed methodology involves denoising and enhancing the tomato leaf images using Binary Wavelet Transform combined with Retinex (BWTR) technique. • The tomato leaves are separated from the background using KSW optimization by the Artificial Bee Colony algorithm (ABCK) for accurate disease recognition. • The Both-channel Residual Attention Network model (B-ARNet) is employed to classify the diseases in the segmented tomato leaf images, achieving an overall detection accuracy of approximately 89
Software Tools (What program/software is used for design, coding and simulation?)	Tensorflow,Python,Anaconda Notebook
Test / Experiment How to test and characterize the design/prototype?	The design/prototype can be tested and characterized by evaluating its performance using evaluation metrics such as accuracy, precision, recall, and F1-score on a test dataset of tomato leaf images, comparing it with other neural network models and analyzing the recognition results and curves.
Simulation/Test Data (What parameters are determined?)	<p>The parameters determined for the simulation/test data include the date of data acquisition (June 8, 2019), location (Hunan Vegetable Institute), camera resolution (4460x3740), number of original images (8616), image pre-processing techniques (denoising, segmentation, random rotation, translation, scaling), and the number and proportion of each tomato leaf disease category (early blight, late blight, leaf mold, bacterial leaf spot, yellow leaf curl disease).</p>  <p>(a)early blight, (b)Citrinitas Leaf Curl,(c) late blight, (d)leaf mold, (e) bacterial leaf spot</p> <p>Fig. 2. Common tomato leaf diseases.</p>
Result / Conclusion (What was the final result?)	The proposed tomato leaf disease recognition method based on ABCK-BWTR and B-ARNet achieves an overall detection accuracy of approximately 89% on 8616 images, demonstrating its effectiveness.
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Future work should focus on enriching the image data of tomato leaf diseases and improving the model's generalization ability.
Terminology (List the common basic words frequently used in this research field)	machine vision technology, noise, image acquisition, disease recognition, Binary Wavelet Transform (BWTR), Retinex.

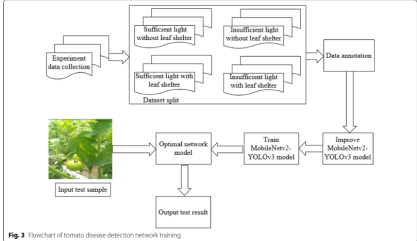
<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed techniques achieves an overall detection accuracy of approximately 89% on 8616 images, • Shengyi Zhao's The SE-ResNet50 model achieved an average detection accuracy of 96.81% for the 10-class classification of tomato leaf diseases
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>Based on the review outcome, the obtained knowledge will be utilized to develop a novel methodology integrating machine vision technology, noise reduction techniques, and the use of Binary Wavelet Transform (BWTR) and Retinex algorithms for disease recognition in my own research project.</p>

Aspects	Paper # 6 (Title)																						
Title / Question (What is problem statement?)	A Smart Solution for Tomato Leaf Disease Classification by Modified Recurrent Neural Network with Severity Computation																						
Objectives / Goal (What is looking for?)	The objective is to design and develop an intelligent tomato leaf disease classification model using artificial intelligence-assisted deep learning approaches.																						
Methodology / Theory (How to find the solution?)	<p>The methodology involves data collection, pre-processing, spot segmentation, deep feature extraction, feature selection, disease classification, and severity computation to develop an accurate tomato leaf disease classification model using AI-assisted deep learning approaches.</p>  <p>Figure 1: The architecture of tomato leaf disease classification and severity computation using</p>																						
Software Tools (What program/software is used for design, coding and simulation?)	Python, Google Colab																						
Test / Experiment How to test and characterize the design/prototype?	The design/prototype can be tested and characterized by evaluating its performance using evaluation metrics such as accuracy, precision, recall, and F1-score on a test dataset of tomato leaf images, comparing it with other neural network models and analyzing the recognition results and curves.																						
Simulation/Test Data (What parameters are determined?)	<p>The parameters determined for simulation/testing include accuracy, sensitivity, specificity, precision, and disease severity computation.</p> <p>Table 2. Examination of severity computation for the designed tomato leaf disease classification.</p> <table> <thead> <tr> <th>Classified outcomes</th><th>Severity score</th></tr> </thead> <tbody> <tr> <td>"Tomato yellow leaf curl virus"</td><td>23.133651009757205</td></tr> <tr> <td>Septoria leaf spot</td><td>23.337791776329983</td></tr> <tr> <td>Septoria leaf spot</td><td>29.95147705078125</td></tr> <tr> <td>Septoria leaf spot</td><td>4.117994349091216</td></tr> <tr> <td>"Tomato yellow leaf curl virus"</td><td>20.336357601239214</td></tr> <tr> <td>Fungal leaf Early blight</td><td>46.478873239436616</td></tr> <tr> <td>Fungal leaf Early blight</td><td>87.08425949857843</td></tr> <tr> <td>Pest Leaf serpentine miner</td><td>78.59237536656892</td></tr> <tr> <td>Pest Leaf serpentine miner</td><td>43.83022640779204</td></tr> <tr> <td>Pest Leaf serpentine miner</td><td>75.83050346965894</td></tr> </tbody> </table>	Classified outcomes	Severity score	"Tomato yellow leaf curl virus"	23.133651009757205	Septoria leaf spot	23.337791776329983	Septoria leaf spot	29.95147705078125	Septoria leaf spot	4.117994349091216	"Tomato yellow leaf curl virus"	20.336357601239214	Fungal leaf Early blight	46.478873239436616	Fungal leaf Early blight	87.08425949857843	Pest Leaf serpentine miner	78.59237536656892	Pest Leaf serpentine miner	43.83022640779204	Pest Leaf serpentine miner	75.83050346965894
Classified outcomes	Severity score																						
"Tomato yellow leaf curl virus"	23.133651009757205																						
Septoria leaf spot	23.337791776329983																						
Septoria leaf spot	29.95147705078125																						
Septoria leaf spot	4.117994349091216																						
"Tomato yellow leaf curl virus"	20.336357601239214																						
Fungal leaf Early blight	46.478873239436616																						
Fungal leaf Early blight	87.08425949857843																						
Pest Leaf serpentine miner	78.59237536656892																						
Pest Leaf serpentine miner	43.83022640779204																						
Pest Leaf serpentine miner	75.83050346965894																						
Result / Conclusion (What was the final result?)	The final result showed that the proposed SD-GHHO-MRNN model achieved superior performance in classifying tomato leaf diseases, with higher accuracy and sensitivity compared to other existing approaches. The severity computation provided accurate classification and quantification of disease spots, enabling effective decision-making in disease management.																						
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Future work includes developing occlusion invariant tomato leaf classification models under real-world scenarios using intelligent approaches like hybrid or ensemble deep learning models.																						
Terminology (List the common basic words frequently used in this research field)	Convolutional neural network; modified recurrent neural network; optimized K-means clustering; residual networks; severity computation; standard deviation-based grasshopper horse herd optimization;																						

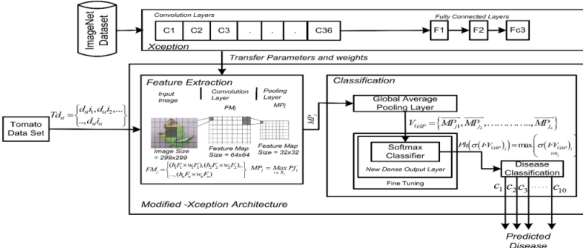
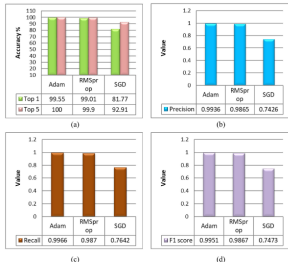
<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed techniques achieves ccuracy” 92.856% on dataset 1 92.795% in dataset 2 and 94.365% in dataset 3 • Surampalli Ashok’s techniques achieved the accuracy 98.12% in tomato leaf disease identification
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>The obtained knowledge will be used to inform and inspire the development of a new methodology for a separate research project focused on enhancing the accuracy and efficiency of disease classification in agricultural systems.</p>

Aspects	Paper # 7 (Title)																																																
Title / Question (What is problem statement?)	Tomato plant disease detection using transfer learning with C-GAN synthetic images																																																
Objectives / Goal (What is looking for?)	The objective is to develop a deep learning-based method for early detection and classification of tomato plant diseases using synthetic and real images.																																																
Methodology / Theory (How to find the solution?)	The proposed methodology utilizes Conditional Generative Adversarial Network (C-GAN) to generate synthetic images of tomato plant leaves, which are then used along with real images to train a DenseNet121 model using transfer learning for accurate classification of tomato leaf diseases.																																																
Software Tools (What program/software is used for design, coding and simulation?)	Python,Google Colab																																																
Test / Experiment How to test and characterize the design/prototype?	The proposed method can be tested and characterized by evaluating its performance in tomato disease detection and classification using the PlantVillage dataset. The accuracy of the model can be measured by comparing the predicted classes of the tomato leaf images with the ground truth labels. Additionally, performance metrics such as precision, recall, and F1-score can be calculated to assess the model's effectiveness in detecting and classifying different diseases.																																																
Simulation/Test Data (What parameters are determined?)	<p>The parameters that are determined in the simulation/test data include the synthetic images generated by the C-GAN model for data augmentation, the real images of tomato plant leaves from the PlantVillage dataset, and the corresponding ground truth labels for disease classification.</p> <p>Precision, Recall and F1-score for different disease classes of tomato plant.</p> <table><tr><th>Class</th><th>Precision</th><th>Recall</th><th>F1-score</th></tr><tr><td>Yellow leaf curl virus</td><td>0.97</td><td>0.97</td><td>0.97</td></tr><tr><td>Septoria leaf spot</td><td>0.97</td><td>0.98</td><td>0.97</td></tr><tr><td>Two spotted spider mite</td><td>1.00</td><td>0.97</td><td>0.98</td></tr><tr><td>Bacterial spot</td><td>1.00</td><td>0.97</td><td>0.98</td></tr><tr><td>Leaf mold</td><td>0.96</td><td>0.97</td><td>0.96</td></tr><tr><td>Mosaic virus</td><td>0.89</td><td>0.98</td><td>0.93</td></tr><tr><td>Target spot</td><td>1.00</td><td>0.99</td><td>0.99</td></tr><tr><td>Early blight</td><td>0.94</td><td>0.99</td><td>0.97</td></tr><tr><td>Late blight</td><td>0.99</td><td>1.00</td><td>0.99</td></tr><tr><td>Healthy</td><td>0.95</td><td>0.87</td><td>0.91</td></tr><tr><td>Mean</td><td>0.97</td><td>0.97</td><td>0.97</td></tr></table>	Class	Precision	Recall	F1-score	Yellow leaf curl virus	0.97	0.97	0.97	Septoria leaf spot	0.97	0.98	0.97	Two spotted spider mite	1.00	0.97	0.98	Bacterial spot	1.00	0.97	0.98	Leaf mold	0.96	0.97	0.96	Mosaic virus	0.89	0.98	0.93	Target spot	1.00	0.99	0.99	Early blight	0.94	0.99	0.97	Late blight	0.99	1.00	0.99	Healthy	0.95	0.87	0.91	Mean	0.97	0.97	0.97
Class	Precision	Recall	F1-score																																														
Yellow leaf curl virus	0.97	0.97	0.97																																														
Septoria leaf spot	0.97	0.98	0.97																																														
Two spotted spider mite	1.00	0.97	0.98																																														
Bacterial spot	1.00	0.97	0.98																																														
Leaf mold	0.96	0.97	0.96																																														
Mosaic virus	0.89	0.98	0.93																																														
Target spot	1.00	0.99	0.99																																														
Early blight	0.94	0.99	0.97																																														
Late blight	0.99	1.00	0.99																																														
Healthy	0.95	0.87	0.91																																														
Mean	0.97	0.97	0.97																																														
Result / Conclusion (What was the final result?)	The proposed deep learning-based method achieved high accuracy rates of 99.51%, 98.65%, and 97.11% for classifying tomato leaf images into 5, 7, and 10 disease categories, respectively, outperforming existing methodologies.																																																
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	The future work aims to extend the proposed method for disease identification and classification in various parts of the plant, such as fruits, stems, and branches, as well as identify different phases of plant diseases.																																																
Terminology (List the common basic words frequently used in this research field)	Deep learning Tomato plant disease detection Conditional Generative Adversarial Network Data augmentation Pre-trained DesnseNet121 network Synthetic Images																																																

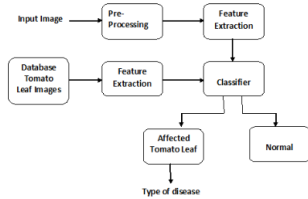
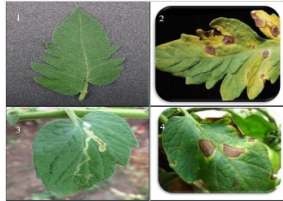
<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed techniques achieves ccuracy” 92.856% on dataset 1 92.795% in dataset 2 and 94.365% in dataset 3 • Jun Liu’s techniques achieved a high F1 score of 93.24% and AP value of 91.32% for tomato gray leaf spot detection
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>The obtained knowledge from the proposed deep learning-based method for tomato plant disease detection can be used as a reference to develop a separate and new methodology for my own research project on plant disease detection and classification.</p>

Aspects	Paper # 8 (Title)
Title / Question (What is problem statement?)	Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model
Objectives / Goal (What is looking for?)	The objective of this study is to develop a mobile app using deep learning techniques for real-time detection and early recognition of tomato gray leaf spot disease.
Methodology / Theory (How to find the solution?)	<p>The proposed method utilizes the MobileNetv2-YOLOv3 model, incorporating the GIoU bounding box regression loss function and a pre-training approach combining mixup training and transfer learning, to achieve accurate and real-time detection of tomato gray leaf spot disease.</p>  <p><small>Fig. 3 Flowchart of tomato disease detection network training</small></p>
Software Tools (What program/software is used for design, coding and simulation?)	Python, Google Colab, tensorflow
Test / Experiment How to test and characterize the design/prototype?	The research utilized a dataset consisting of 2,385 images of tomato gray leaf spot captured in the Shouguang tomato planting base in Shandong Province, China. The dataset was randomly divided into training, validation, and test sets. The images were annotated using the LabelImg tool and stored in the PASCAL VOC dataset format. The MobileNetv2-YOLOv3 model was used for object detection, which combines the MobileNetv2 backbone network with the YOLOv3 algorithm. The loss function used for bounding box regression was based on Generalized Intersection over Union (GIoU).
Simulation/Test Data (What parameters are determined?)	The parameters determined for simulation/test data include meteorological conditions, lighting conditions (sunny, cloudy, rainy), presence or absence of leaf shelter, and the number and types of tomato gray leaf spot images.
Result / Conclusion (What was the final result?)	Future work could involve expanding the detection system to include other common diseases in tomatoes, developing computer or mobile app platforms for practical use, and integrating multiple data sources for an early warning model of tomato gray leaf spot disease.
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	
Terminology (List the common basic words frequently used in this research field)	YOLOv3, Convolutional neural network, Tomato diseases and pests, Object detection

<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed techniques achieved a high F1 score of 93.24% and AP value of 91.32% for tomato gray leaf spot detection • Shengyi Zhao's The SE-ResNet50 model achieved an average detection accuracy of 96.81% for the 10-class classification of tomato leaf disease
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>The obtained knowledge will be used to develop a novel methodology for plant disease identification in my own research project, incorporating techniques such as deep learning models and an expanded dataset for improved accuracy.</p>

Aspects	Paper # 9 (Title)
Title / Question (What is problem statement?)	Automated tomato leaf disease classification using transfer learning-based deep convolution neural network
Objectives / Goal (What is looking for?)	The objective of this study is to propose a transfer learning-based deep convolutional neural network model for accurate detection and classification of tomato leaf diseases using minimal plant image data.
Methodology / Theory (How to find the solution?)	<p>A Modified-Xception model was trained using a dataset of tomato leaf images, employing specific hardware and deep learning frameworks. The model underwent feature extraction, classification, and fine-tuning processes with different optimization algorithms, achieving accurate disease classification.</p> 
Software Tools (What program/software is used for design, coding and simulation?)	Python, Google Colab, tensorflow
Test / Experiment How to test and characterize the design/prototype?	The design/prototype will be tested by feeding it with real-time and stored tomato leaf images, and its performance will be evaluated using evaluation metrics such as accuracy, precision, recall, and F1-score.
Simulation/Test Data (What parameters are determined?)	The parameters determined in the simulation/test data for this research project are the accuracy of different models in tomato leaf disease identification using various deep learning architectures (AlexNet, GoogLeNet, SqueezeNet, ResNet, VGG16, and Modified-Xception) on different dataset sizes and number of classes.
Result / Conclusion (What was the final result?)	<p>Modified-Xception model achieved highest accuracy</p>  <p>99.55%.</p>
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Future work could focus on expanding the dataset and incorporating additional crop categories for improved accuracy in plant disease identification.
Terminology (List the common basic words frequently used in this research field)	Deep convolution neural network · Tomato leaf disease · Transfer learning · Feature extraction · Disease classification · Fine-tuning

<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed techniques achieved the highest accuracy of 99.55% in tomato leaf disease identification • Shengyi Zhao's The SE-ResNet50 model achieved an average detection accuracy of 96.81% for the 10-class classification of tomato leaf disease
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>The outcome of the study suggests that incorporating the Modified-Xception model trained with Adam optimizer and evaluated using appropriate performance metrics can serve as a valuable methodology for tomato disease classification in my own research project.</p>

Aspects	Paper # 10 (Title)
Title / Question (What is problem statement?)	Tomato Leaf Disease Detection Using Deep Learning Techniques
Objectives / Goal (What is looking for?)	The objective of this paper is to develop a reliable and accurate system for early detection of tomato plant leaf diseases using image processing techniques.
Methodology / Theory (How to find the solution?)	<p>The proposed methodology involves image processing techniques, including preprocessing with a Gaussian filter, feature extraction using Discrete Wavelet Transform (DWT) and Gray-Level Co-occurrence Matrix (GLCM), and segmentation using color-based thresholding, to classify tomato leaf diseases and suggest suitable solutions.</p>  <pre> graph LR Input[Input Image] --> Pre[Pre-Processing] Pre --> FE1[Feature Extraction] FE1 --> Classifier[Classifier] DB[Database Tomato Leaf Images] --> FE2[Feature Extraction] FE2 --> Classifier Classifier --> Affected[Affected Tomato Leaf] Classifier --> Normal[Normal] Affected --> Disease[Type of disease] </pre> <p><i>Figure 7. The overall framework for Tomato Leaf Disease Detection method</i></p>
Software Tools (What program/software is used for design, coding and simulation?)	Python, Google Colab, tensorflow
Test / Experiment How to test and characterize the design/prototype?	The proposed method was tested using OpenCV and evaluated by comparing it with other methods, including Deep Neural Networks (AlexNet) and ANN technique. The performance measures, including accuracy rates, were calculated to assess the effectiveness of the proposed method in detecting tomato leaf diseases.
Simulation/Test Data (What parameters are determined?)	<p>The simulation/test data involved tomato leaf images with different diseases and healthy leaves, and the parameters determined were the accuracy rates for disease detection and classification.</p>  <p><i>Figure 2. Sample Tomato leaf images with four disease conditions. 1. Healthy 2. Phoma Rot 3. Leaf Miner 4. Target Spot.</i></p>
Result / Conclusion (What was the final result?)	The proposed method achieved high accuracy rates of 98.12% in detecting and classifying tomato leaf diseases, outperforming other methods such as Deep Neural Networks (AlexNet) and ANN technique.
Obstacles/Challenges (List the methodological obstacles if authors mentioned in the article)	Future work could involve exploring the use of additional algorithms and techniques, such as artificial neural networks, fuzzy logic, and hybrid approaches, to further improve the accuracy and effectiveness of tomato leaf disease detection.
Terminology (List the common basic words frequently used in this research field)	Leaf disease detection, image processing, convolutional neural networks, feature extraction, deep learning

<p>Review Judgment (Briefly compare the objectives and results of all the articles you reviewed)</p>	<ul style="list-style-type: none"> • The reviewed techniques achieved the accuracy 98.12% in tomato leaf disease identification • Rajasekaran Thangaraj's The Modified Xception model achieved an average detection accuracy of 99.55% in tomato leaf disease detection.
<p>Review Outcome (Make a decision how to use/refer the obtained knowledge to prepare a separate and new methodology for your own research project)</p>	<p>The obtained knowledge from the reviewed paper can be used as a reference to inform and inspire the development of a new methodology for a separate research project on plant disease detection, tailored to the specific objectives and requirements of the project.</p>