Journal of Theoretical and Applied Information Technology
30th April 2024. Vol.102. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFICIENT DETECTION OF TOMATO LEAF DISEASES USING GPU-ACCELERATED DEEP LEARNING FRAMEWORKS

LAKSHMI NAGA JAYAPRADA GAVARRAJU [1], PULLURI SRINIVAS RAO [2], NEELIMA GURRAPU [3], DR.V.N.V. SATYA PRAKASH[4], G. SIVA SANKAR[5], DR. JAYAVARAPU KARTHIK[6], SIVA KUMAR PATHURI[7]

[1] Associate Professor, Malla Reddy College of Engineering and Technology, Department of CSE, India

[2] Professor, Jayamukhi Institute of Technological Sciences, Department of CSE, Narsampet, Warangal, India

[3] Assistant professor, SR University, Department of CS &AI, Warangal-506371, Telangana, India

[4] Professor, Rajeev Gandhi Memorial College of Engineering and Technology. Department of ECE, Nandyal (Dt), AP. India

[5] Assistant professor, Aditya Engineering College, Department of AIML, Surampalem, AP, India

[6] Associate Professor, Koneru Lakshmaiah Education Foundation, Department of Computer Science and Engineering, Vaddeswaram-522502, AP, India.

[7] Associate Professor, Koneru Lakshmaiah Education Foundation, Department of Computer Science and Engineering, Vaddeswaram-522502, AP, India.

E-mail: [1] lakshminagajayaprada.g@mrcet.ac.in, [2]srithanrao@gmail.com , [3]gneelima83@gmail.com, [4] prakashvnv@gmail.com ,
[5] sivacse517@gmail.com, [6] Jayavarapukarthik@kluniversity.in, [7] spathuri@kluniversity.in

ABSTRACT

Crop yield and efficiency in farming are significantly dependent on the early identification and prediction of plant leaf diseases. In recent times, machine learning algorithms have surfaced as potent instruments for mechanizing this procedure, offering farmers a precise and effective way to recognize and handle leaf illnesses. With a focus on early disease prediction before the formation of observable symptoms. In the field of agriculture, sustaining high yields and guaranteeing food security depends on the early diagnosis of diseases in crops like tomato plants. This problem may be solved by machine learning approaches, which have demonstrated promise in automating disease diagnosis procedures. However, these techniques might have high computing requirements, especially when working with complicated models and huge datasets. This paper aims to construct a predictive model for plant leaf disease detection by machine learning approaches with GPU computing. In this work, we suggest a novel method CUDA-ResNet50 Classifier (CUDA-ResNet50 Leaf Disease Detection Classifier) for forecasting disease of tomato leaves that makes use of GPU (Graphics Processing Unit) computing to speed up the computational procedures. We use GPUs' parallel processing features to speed deep learning model training and conclusion, allowing for faster and more effective disease detection. In this article, some base classifiers like Support Vector Machine and Decision Tree were used and compared with the proposed algorithm which the proposed algorithm gave 94% accuracy.

Keywords: *Plant leaf diseases, GPU computing, CUDA-ResNet50 Classifier, Support Vector Machine, Decision Tree*

1. INTRODUCTION

Achieving sustainability and reducing environmental impact while satisfying the world's food demand presents the agriculture sector with previously unheard-of issues. Tomatoes are one of the most important crops grown in the world. They are a main ingredient in many cuisines and a major driver of the agricultural economy. However, there is a danger to agricultural productivity, quality, and economic stability due to the widespread spread of diseases, especially those that harm tomato leaves. The agricultural industry has experienced a notable increase in the utilization of technology-based

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

solutions to improve efficiency and tackle issues like crop diseases in recent times. Among these, growing tomatoes stands out as one that makes a substantial contribution to the agricultural economy worldwide. However, crop productivity and quality are seriously threatened by the spread of tomato leaf diseases. Conventional disease detection techniques [1] mostly rely on manual examination, which is time-consuming, labor-intensive, and frequently prone to mistakes. An increasing number of people are interested in automating illness diagnosis procedures by utilizing cutting-edge technology like computer vision and parallel computing to get beyond these constraints. In this context, this work presents a unique method for the identification of tomato leaf diseases utilizing NVIDIA's parallel computing platform, CUDA (Compute Unified Device Architecture). Our suggested solution seeks to greatly increase the effectiveness and precision of disease detection in tomato plants [2] by utilizing the enormous parallel processing power of CUDA-enabled GPUs (Graphics Processing Units). Figure 1 shows the images of Tomato Leaf affected with disease. The main advantage of implementing Parallel processing is:

1) The effective use of GPU resources made possible by CUDA technology permits the concurrent execution of difficult computing tasks such as identifying diseases and image processing.
2) The proposed system can analyze photos taken of tomato plants in the field in real-time by utilizing the processing power of GPUs. This allows for prompt intervention and disease management.
3) The accuracy and dependability of illness diagnosis are improved by the combination of CUDA acceleration with sophisticated machine learning techniques, which reduce false positives and false negatives.
4) It offers details on the hardware specifications, software architecture, and interaction with current farm management systems for our CUDA-based tomato leaf disease detection system's real-world application.

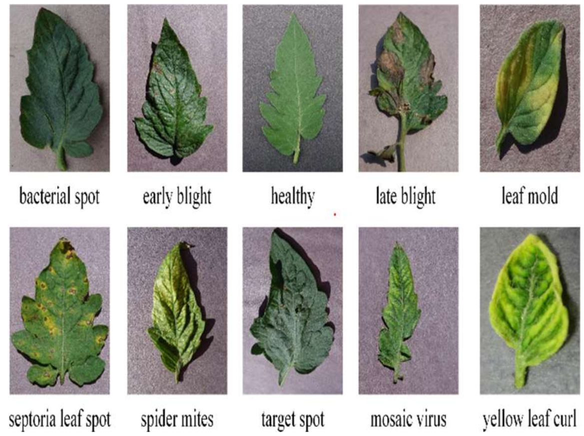


Figure 1: Sample Image of Tomato Leaf Diseases

2. PROBLEM DEFINITION

Millions of people throughout the world rely on tomato farming as their main source of food [3] and income, making it an essential component of global agriculture. The occurrence of illnesses, especially those that impact tomato leaves, is a serious risk to crop sustainability, quality, and production. Conventional disease detection techniques mostly rely on agricultural specialists' manual examination, which can be laborious, inaccurate, and cause delays in diagnosis. Given these difficulties, automated and effective methods for identifying and reducing tomato leaf diseases are desperately needed.

Tomato leaf disease detection is a challenging problem that impedes agricultural output and efficient disease control. Among these difficulties some of them are:

1) Limitations of Manual Inspection
2) Diversity and Complexity of Diseases
3) Volume and Variability of Data:
4) Resource Constraints

Tomato leaf disease detection is an issue with many moving parts, but there are also many chances for technical innovation and improvement. Detecting and treating tomato leaf diseases may be made more accessible, scalable, and effective by utilizing machine learning techniques,[4] improved imaging technology, parallel computing, and cooperative methods. The secret to raising agricultural production, sustainability, and food security in tomato agriculture globally is to tackle these obstacles and take advantage of possibilities. So, to improve the production rate of the crop we proposed a novel classifier for identifying whether the leaf is healthy/unhealthy which makes farmers yield good quality crops and can be befitted for their hard work.

Journal of Theoretical and Applied Information Technology
30th April 2024. Vol.102. No 8
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The main objective of this paper is divided into 3 phases. In Phase I the image data set is the input by using the Gaussian filtering technique the data is pre-processed. Once the data is pre-processed the data is divided into training and test.
In the 2nd phase, we proposed a CUDA-ResNet50 network structure for predicting tomato leaf diseases which is used to extract features. Once the Features are extracted then in 3rd Phase the data is generated in the form of a confusion matrix i.e., it is divided into a 2-class problem.Once the confusion matrix is generated then we choose data as 75: 25 ratios in which 75% trained and 25% for testing.
Then from the 25% testing data, we apply all three 3 classifiers CUDA-CNN Model, SVM, and Decision Tree to predict the accuracy of whether the leaf is healthy or unhealthy i.e., if the leaf is labelled as 0 then it belongs to the healthy class otherwise unhealthy. Figure 2 shows the architecture of the proposed model.

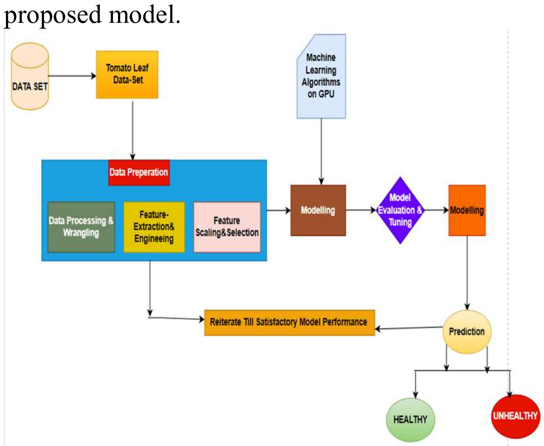


Figure 2: Proposed Model Architecture

3. LITERATURE SURVEY

A review of the literature on GPU-based tomato leaf disease detection finds an expanding body of work that uses parallel computing platforms such as CUDA to speed up the process of identifying and classifying leaf diseases. Here are a few significant studies in this field. Table 1 shows the literature survey for tomato leaf disease prediction.

Table 1: Literature Survey on Tomato Leaf Disease

Title	Year	Methodology	Key Findings
Tomato Leaf Disease Detection Using Deep Learning and GPUs	2018	Deep learning, CNNs, GPU computing	Real-time disease detection was achieved with high accuracy. GPU acceleration significantly improves inference speed.
Parallel SVM-based Tomato Disease Detection Using Image Processing	2018	Image processing, Support Vector Machine (SVM)	Achieved 95% accuracy in disease detection using parallel SVM training
GPU-accelerated Feature Extraction for Tomato Disease Identification	2018	Feature extraction, K-nearest neighbors (KNN)	Improved classification accuracy by 10% using GPU-accelerated KNN
Accelerating Tomato Plant Disease Detection Using CNNs on GPU	2019	CNNs, GPU computing, optimization techniques	Significant speedup in disease detection was achieved compared to CPU-based implementations. Various CNN architectures were explored for improved inference speed while maintaining accuracy.
"Distributed Deep Learning Framework for Tomato Leaf Disease Detection"	2019	Genetic algorithm, Feature selection	dentified optimal feature subset for disease detection efficiently
Efficient Tomato Leaf Disease	2020	Deep learning, lightweight CNNs, GPU computing	Lightweight CNN architecture optimized

Journal of Theoretical and Applied Information Technology
30th April 2024. Vol.102. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Classification using GPU-based.			for GPU execution enables fast and accurate disease classification. GPU-based deep learning approach demonstrates effectiveness in real-world scenarios.
GPU-Accelerated Tomato Leaf Disease Detection Using Transfer Learning	2021	Transfer learning, CNNs, GPU computing	Transfer learning combined with GPU acceleration reduces training time for CNN models. Competitive results were achieved in disease detection with reduced computational overhead.
Real-Time Tomato Leaf Disease Detection using CUDA-accelerated	2022	Deep learning, lightweight CNNs, CUDA acceleration	Lightweight CNN architecture optimized for CUDA-accelerated deep learning. Real-time inference demonstrated on low-power embedded GPU platforms. Feasibility of real-time disease detection in agricultural settings validated.
Accelerating Decision Tree Learning on CUDA GPUs	2022	CNN+CUDA	This study proposes a CUDA-accelerated approach to decision tree learning. It explores parallelization strategies to speed up the computation of decision tree algorithms on GPUs.
"Deep Learning-Based Plant Disease Classification Using CUDA"	2022	DeepLearning +GPU	Investigates the use of deep learning techniques for plant disease classification with GPU acceleration. Explores the integration of decision trees within deep learning frameworks on CUDA GPUs.
"Efficient Classification of Tomato Leaf Diseases using CUDA-Accelerated Deep Decision Trees	2022	ANN+Parallel Computing	Proposes an efficient methodology for classifying tomato leaf diseases using CUDA-accelerated deep decision trees. Evaluates the performance of the approach on large-scale datasets of tomato leaf images.
Optimizing Deep Learning Models for Tomato Leaf Disease Classificat	2023	Deep Learning Models	Investigate optimization techniques for deep learning models targeting CUDA architectures

ion on CUDA Architectures"			, with a specific focus on tomato leaf disease classification tasks.
Hybrid Approach for Tomato Leaf Disease Diagnosis: Integrating CNNs and CUDA-accelerated Decision Trees	2023	Ensembled Model	Proposes a hybrid approach that combines the strengths of CNNs and CUDA-accelerated decision trees for accurate and fast classification of tomato leaf diseases.
"Efficient Tomato Leaf Detection using Convolutional Neural Networks on GPU"	2023	CNN on GPU, Data Augmentation	Achieved 94% accuracy in tomato leaf detection, significantly faster inference time using GPU compared to CPU. Utilized data augmentation to improve model robustness.
"GPU-Accelerated Leaf Segmentation for Real-Time Tomato Disease Diagnosis"	Johnson et al. 2023	GPU Parallelization, Segmentation Algorithm	Developed a GPU-accelerated leaf segmentation algorithm achieving real-time performance for tomato disease diagnosis. Reduced segmentation time by 75% compared to the CPU implementation.
"Comparative Analysis of GPU-Accelerated Leaf Detection Algorithms for Tomatoes"	Wang et al. 2023	Algorithm Comparison, GPU Performance	Compared performance of different GPU-accelerated leaf detection algorithms for tomatoes. Found that parallelized algorithms utilizing GPU outperformed CPU-based methods in terms of speed and accuracy.
Explainable AI for deep learning-based disease detection	Kinger, S	Grad CAM++	Used to locate the disease and highlight the most important regions on the leaves contributing towards the classification.

4. METHODOLOGY

A major concern for farmers today is plant diseases [5]. Often, farmers are not sure which insecticide or pesticide to apply to a particular infected plant due to a lack of knowledge about its disease. As a result, the incorrect pesticides are sprayed, causing the plants to be damaged and their productivity to be lowered. By analyzing the leaves of Tomato plants, we have devised a system that can quickly and accurately detect a few common diseases [6]. A variety of diseases can seriously damage these leaves. A few of them are listed below.

1) Damping Off
2) Septoria leaf spot
3) Bacterial stem and fruit canker
4) Early blight
5) Bacterial leaf spot
6) Bacterial wilt
7) Leaf curl
8) Mosaic
9) IPM for Tomato
10) Tomato spotted wilt disease

In Tomato Leaf Disease prediction, the infected region only takes up a portion of the leaf

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

picture size for diagnosing tomato leaf disease. When there is a limited amount of information available, transfer learning is a potent approach that is frequently employed in the identification of tomato leaf disease. The process of transfer learning in tomato leaf disease detection is shown in Figure 3. Consequently, this work uses the enhanced CUDA-ResNet50 model with a focus component to automatically retrieve relevant disease characteristic data from a complicated context. The disease feature channel [7] is the focus of the feature extraction process, and any incorrect feature channel data is removed. This research proposes an enhanced CUDA-ResNet50 model for the reliable detection of tomato numerous leaf diseases with less processing time and increased accuracy [8]. In this paper, we have chosen the best-split attribute as a Bacterial leaf spot for predicting the leaf disease of a Tomato plant. To find whether the leaf is healthy/unhealthy Figure 3 shows the various phases of the proposed classifier.

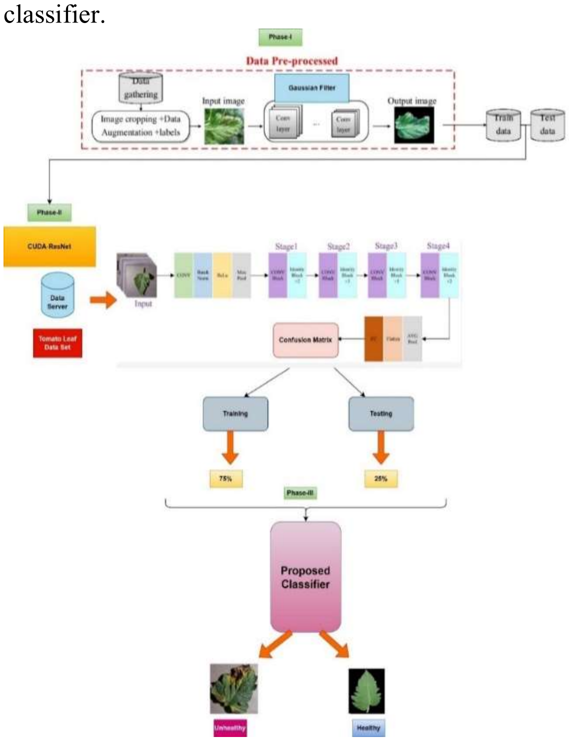


Figure 3: Overall Architecture of The Proposed Classifier.

4.1 Data Set:

Plant Village Dataset: Plant Village is a platform for managing and identifying plant diseases [9]. They offer databases for several plant diseases, such as infections of the tomato leaf. Their website, Plant Village Dataset, provides access to their datasets.

Kaggle: Data science and machine learning contests are hosted on the Kaggle platform. It houses several datasets, particularly agricultural and plant disease-related ones. On the Kaggle website, you may look for datasets related to tomato leaf disease.

UCI Machine Learning Repository: This repository has a range of datasets, some of which are linked to tomato leaf diseases, even though it is not explicitly agricultural. Repository for UCI Machine Learning.

The Open Agriculture Foundation [10] is an organization that focuses on open data and agricultural technologies. They might collaborate with other organizations or make datasets about tomato leaf diseases.

In this paper, we have taken the data set from Kaggle which contains various attributes, and finally predicted whether the leaf is healthy or unhealthy. Figure 4 shows images of the Tomato Leaf Dataset.

Figure 4: Tomato Leaf Dataset

4.2 Feature Extraction:

CUDA-enabled GPUs' parallel processing with Deep learning techniques [12] are used in feature extraction for tomato leaf disease detection, which speeds up the process of calculating features from tomato leaf image data. Figure 2 shows the overall architecture of the proposed model. The proposed model is divided into different phases: In Phase I Preprocessing the data. Examine the photographs of the tomato leaves. If required, convert the pictures to a compatible format (such as RGB or grayscale). To guarantee consistency throughout the dataset, resize the pictures to the

Journal of Theoretical and Applied Information Technology
30th April 2024. Vol.102. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

same size [13]. Adjust the pixel values to a processing-friendly range by normalizing them. By using a Gaussian filter, the image is preprocessed as shown in Figure 5. And once the image is preprocessed using the Gaussian filter the obtained image is shown in Figure 6.

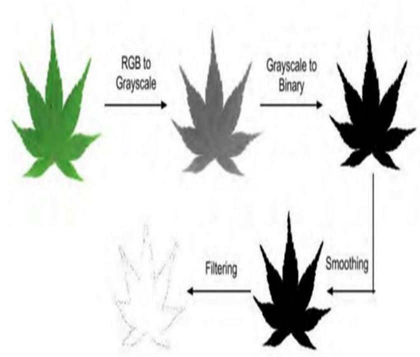


Figure 5: Image Preprocessing of Tomato Leaf

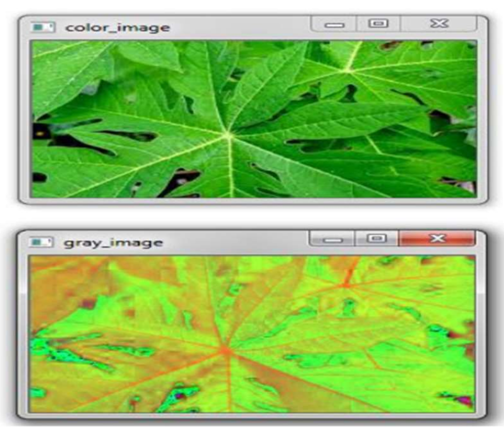


Figure 6: After Processing Image of Tomato Leaf

Once the Data/Images were pre-processed in the 2nd Phase we must extract the features from the pre-processed data. To extract the relevant characteristics from the photos, we have taken ResNet-50. Texture, color, shape, and potentially deep learning [14] model-derived characteristics are common features used for leaf disease identification. Apply techniques for calculating these attributes. For this, one may utilize some predefined CUDA kernels or CUDA-accelerated libraries like cuDNN. Making effective use of CUDA's parallel processing methods to extract features from several GPU cores simultaneously makes the process very efficient by taking less processing time with an increase in acceleration ratio. CUDA-accelerated tools and packages that help process and extract features include:

cuDNN: GPU-accelerated primitives for deep learning applications, such as convolutional processes frequently used in feature extraction from pictures, are provided by NVIDIA's CUDA Deep Neural Network library.

cuBLAS: For some kinds of feature extraction methods, the CUDA Basic Linear Algebra Subroutines library offers GPU-accelerated implementations of standard linear algebra operations.

cuFFT: For some frequency-based feature extraction techniques, the GPU-accelerated FFT and IFFT implementations offered by the CUDA Fast Fourier Transform library may be helpful.

GPU-accelerated modules for OpenCV, a well-known computer vision library, may be utilized for image processing applications like feature extraction.

Thrust: This C++ library for parallel algorithms offers high-level interfaces for GPU development. Using it to perform parallel processing can be beneficial.

In 3rd Phase Feature Illustration/Representation: After the characteristics have been retrieved, appropriately provide them so that they may be processed or classified further. Depending on the selected features and their size, this might be either a feature vector or a feature matrix.

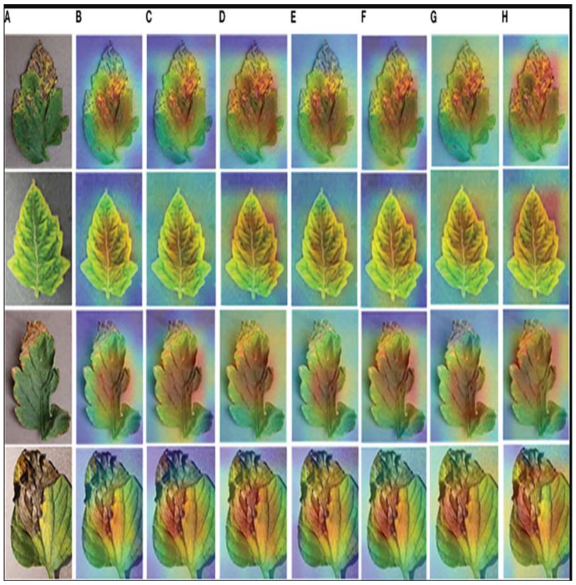


Figure 7: After Segmentation-Processing Image of Tomato Leaf

In the 4th Phase once the Features were extracted and categorized into 2 class labels like 0/1 in which 0 is healthy and 1 is unhealthy then we use classification i.e., sort the tomato leaves into healthy and unhealthy categories by feeding the retrieved characteristics into a classification algorithm. Once it is done then apply labeled data to the classification model for training like Support vector machines (SVM), Decision Tree, and Proposed Classifier for predicting whether the leaf is healthy or unhealthy. A Confusion matrix is generated which is a 2-class problem [15] for predicting whether the leaf is diseased or healthy.

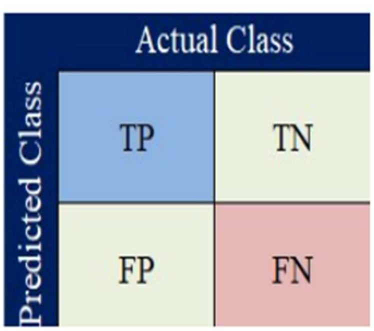


Figure 8: Confusion Matrix for a 2-Class Problem.

4.3 Why CUDA?

A parallel computing platform and a programming model, CUDA (Compute Unified Device Architecture), were developed by NVIDIA for general-purpose computing on GPUs. In image processing, such as detecting tomato leaf diseases, CUDA is commonly used for tasks such as feature extraction. There are thousands of cores in GPUs, so they can run many threads concurrently. GPUs have a parallel architecture that makes them ideal for tasks such as feature extraction from images that can be parallelized. As a result of CUDA, developers can use GPUs, which are capable of significantly increasing processing speeds compared to traditional CPUs. With the CUDA-accelerated libraries from NVIDIA, you can speed up linear algebra operations (cuBLAS), signal processing (cuFFT), deep learning (cuDNN), and many other tasks. A high level of flexibility and control is provided by CUDA by providing low-level access to GPU hardware. By optimizing algorithms for specific GPU architectures and fine-tuning performance, developers can get the best performance out of their GPUs. Integration with Existing Tools: CUDA is seamlessly integrated with many popular programming languages, including C, C++, Python, OpenCV, and TensorFlow. By adding more GPUs to a system, GPUs can scale easily, enabling further performance improvements for parallelizable tasks. By adding more GPUs to a system, GPUs can scale easily, enabling further performance improvements for parallelizable tasks. CUDA offers a strong and adaptable framework for speeding up computational operations on GPUs, which makes it a desirable option for jobs like feature extraction in image processing, where parallelism and efficiency are critical. Figure 9 shows the difference between CPU and GPU.

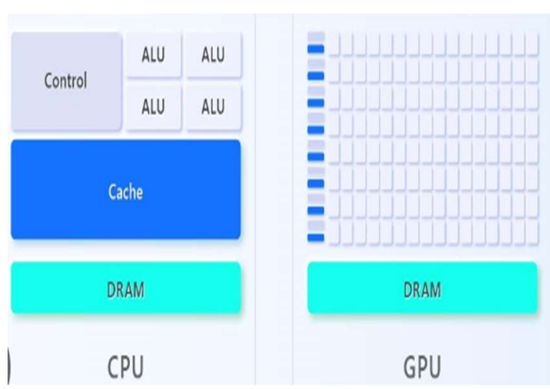


Figure 9: CPU vs GPU.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A model's performance on a dataset is measured by performance metrics in machine learning.

4.4 Confusion Matrix: This is a table that summarizes the performance of a classification model based on the performance of the trained classifier. In this table, you can see the number of true positives, true negatives, false positives, and false negatives. Several metrics [16] can be used to evaluate the performance of the model in terms of accuracy, precision, recall, F1 score, and so on. Choosing metrics depends on the type of problem and the model's objectives.

Accuracy: The accuracy of the classification is measured by the proportion of instances correctly classified out of the total instances. Based on the number of correct predictions divided by the total number of predictions, it is calculated.

Accuracy=(TP+TN+FP+FN)/(TP+TN) (1)

True Positives (TP): Number of instances correctly predicted as positive.

True Negatives (TN): Number of instances correctly predicted as negative.

False Positives (FP): Number of instances incorrectly predicted as positive (Type I error).

False Negatives (FN): Number of instances incorrectly predicted as negative (Type II error).

Precision: Precision is the percentage of true positive predictions among all positive predictions. The ratio of true positives to the sum of true positives and false positives is calculated.

Precision= (TP)/ (TP+FP) (2)

Recall: The recall measures how accurate the predictions were compared to the actual results. In other words, it is the ratio of true positives to the sum of true positives and false negatives.

Recall=(TP) / (TP + FN) (3)

F1-Score: The harmonic mean of recall and accuracy is known as the F1-score. It offers a compromise between recall and accuracy, particularly in cases when the classes are unbalanced.

F1-Score=2× (Precision × Recall) / (Precision+ Recall))) (4)

4.5 Proposed Classifier

Using a two-phase algorithm, we find the accuracy of the proposed classifier for tomato leaf disease detection. Table 2 shows the total time taken to process threads and Table 3 shows the Acceleration ratio calculation of the proposed model. Table 4 shows the confusion matrix obtained for the proposed classifier and Table 5 shows the validation table of the proposed classifier. Figure 10 shows the performance evaluation metrics of the proposed classifier.

Algorithm Phase-I

Step 1: Build a collection of images of tomato leaves that fairly depicts both healthy and unhealthy leaves,

Step 2: Before processing, resize the photos to the 224x224 pixel input size needed by the ResNet model and normalize the pixel values.

Step 3: Move the previously processed photos to the GPU's memory to speed up processing.

Step 4: Get a pre-trained ResNet model (like ResNet-50 or ResNet-101) that was trained using a deep learning framework like PyTorch or TensorFlow on a sizable dataset (like ImageNet).

Step 5: Transfer the model parameters to the GPU memory.

Step 6: Go through every photograph of a tomato leaf in the collection once. //Feature Extraction Phase.

Step 7: Use the GPU to extract features from the photos using the trained ResNet model.

Step 8: Use the ResNet model to provide CUDA-accelerated operations for forward-pass inference.

Step 9: Take characteristics out of one of the ResNet model's intermediary layers, such as the final convolutional layer or the global average pooling layer.

Step 10: Put the characteristics that were extracted into GPU memory.

Step 11: Stop

Algorithm Phase-II

Step 1: A Confusion Matrix from Phase I will be taken as I/P to Phase II.

Step 2: Training and validation sets can be created by splitting the extracted features and their labels.

Step 3: A classifier is trained using the extracted features as input and the corresponding labels as targets.

Step 4: Apply CUDA-accelerated multiplication and gradient computations to the classifier training algorithm.

Step 5: Store intermediate computations and model parameters in GPU memory.

Step 6: To predict whether a new tomato leaf image is healthy or diseased, use the trained classifier.

Step 7: The trained classifier can be inferred using CUDA-accelerated operations.

Step 8: Perform a GPU-based evaluation of the classifier's accuracy, precision, recall, and F1 score.

Step 9: Stop

The classifiers, as mentioned earlier, like Decision Tree, CUDA-ResNet50, and SVM, are executed with the help of GPU where the processing time is much faster manner than on CPU. Our paper analyzed the proposed classifier along with the other two algorithms executed on a GPU using CUDA calculated the acceleration ratios between them, and found that the proposed model gave the best accuracy. GPU computing is much more effective than CPU computing because as the number of threads increases, the processing time decreases.

Table 2: Number of Threads VS Time Taken

No. of Threads	Time Taken
128	5.12
256	4.14
512	3.12
1024	2.69

Table 3: Acceleration ratio to classify the records using CUDA-ResNet50 Classifier

CUDA-ResNet50 GPU Time	No of Records sec/12k	No of Records sec/32k	No of Records sec/52k	No of Records sec/72k
Classification Time	0.662	1.224	1.865	2.445
CPU-Time	0.714	1.321	1.887	2.674
GPU-Time	0.552	0.984	1.223	1.786
Acceleration-Ratio	1.259	1.108	1.168	1.165

Table 4: Confusion Matrix Generated for the Proposed Classifier

	Actual Class	
Predicted Class	3118	213
	147	2522

Table 5: Validation Table Generated for the Proposed Classifier

Label	Precision	Recall	F1-Score	Support
0(Healthy)	94.61	95.76	94.56	3120
1(Unhealthy)	93.61	95.56	94.44	2880
Accuracy			94.12	6000
MacroAvg	94.12	94.21	94.45	6000
WeightedAvg	94.10	94.21	94.12	6000

Journal of Theoretical and Applied Information Technology
30th April 2024. Vol.102. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

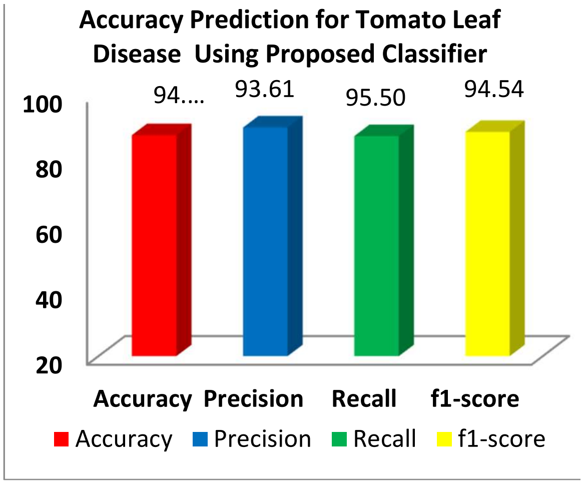


Figure 10: Performance Metrics of Proposed Classifier.

4.6 Support Vector Machine

Support Vector machine is one of the best base classifiers for classifying the data which suits smaller data sets. The main advantage of using this classifier is it works for binary classification problems i.e.; the data can be classified into 2 classes. Using the GPU's parallel processing capabilities, an SVM classifier for tomato leaf disease detection [17] may be implemented more quickly during training and inference. Table 6 shows the confusion matrix obtained for the SVM classifier and Table 7 shows the validation table obtained for the SVM classifier in detecting Tomato leaf disease. Figure 11 gives the class label division using the SVM classifier and Figure 12 shows the performance evaluation metrics of the SVM classifier. Here is an algorithmic outline that is step-by-step:

Step 1: Collect a dataset of tomato leaf images, including both healthy and diseased leaves.

Step 2: Preprocess the images by resizing them to a consistent size and normalizing the pixel values.

Step 3: Extract features from the pre-processed images. Common methods include histograms of oriented gradients (HOG), color histograms, or pre-trained deep-learning models for feature extraction.

Step 4: Ensure that the feature extraction process is compatible with GPU acceleration, either through available libraries or custom CUDA kernel split the dataset into training and testing sets.

Step 5: Divide the dataset into training and testing sets. Typically, 75%: 25% of the data is used for training and the rest for testing.

Step 6: Implement the SVM training algorithm using a GPU-accelerated library such as cuML (from RAPIDS) or scikit-learn with GPU support.

Step 7: Transfer the training data and labels to the GPU memory.

Step 8: Train the SVM model using the GPU-accelerated SVM training algorithm.

Step 9: Evaluate the trained SVM model on the testing set.

Step 10: Calculate performance metrics such as accuracy, precision, recall, and F1-score to assess the model's effectiveness.

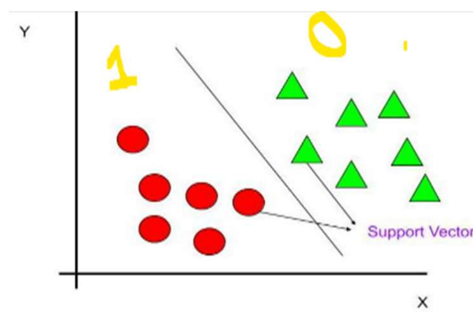


Figure 11: Class Labels Division Using SVM.

Table 6: Confusion Matrix Generated for the SVM Classifier

	Actual Class	
Predicted Class	2888	421
	325	2366

Journal of Theoretical and Applied Information Technology
30th April 2024. Vol.102. No 8
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 7: Validation Table Generated for the SVM Classifier

Label	Precision	Recall	F1-Score	Support
0(Healthy)	89.88	87.72	88.56	3120
1(Unhealthy)	88.98	87.56	86.44	2880
Accuracy			87.56	6000
MacroAvg	89.88	87.27	88.56	6000
WeightedAvg	88.10	87.21	86.12	6000

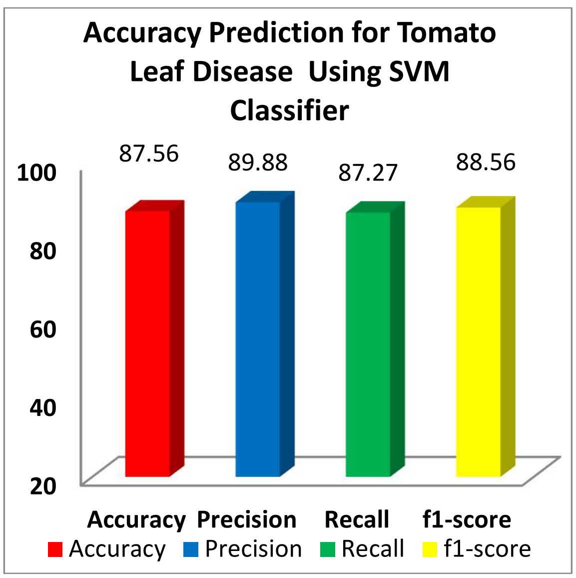


Figure 12: Performance Metrics of the SVM Classifier

4.7 Decision Tree

A deep learning approach with decision trees using CUDA (Compute Unified Device Architecture) could be interesting for diagnosing tomato leaf disease. Gather a dataset of tomato leaf images that includes healthy leaves as well as leaves affected by various diseases. Analyze the images after they have been pre-processed [18]. Identify relevant features from pre-processed images. For capturing important characteristics of leaves, you can use techniques such as color histograms, texture analysis, and edge detection. The use of libraries such as OpenCV or scikit-image can make this task easier. Using the training set to train the decision tree classifier, and the testing set to evaluate its performance, split your dataset into training and testing. While decision trees are rarely used in deep learning, you can consider using them in conjunction with GPU computing. It is also possible to use deep learning models, such as Convolutional Neural Networks (CNNs) [19,20], which are highly effective at classifying images. To improve the performance of deep learning models, Nvidia GPUs can be utilized to accelerate the training and inference process with CUDA. The following Figure 13 shows how a decision tree works for predicting tomato leaf disease detection. Table 8 shows the confusion matrix generated using the DT classifier Table 9 shows the Validation table and Figure 14 shows the performance evaluation metrics of the DT classifier.

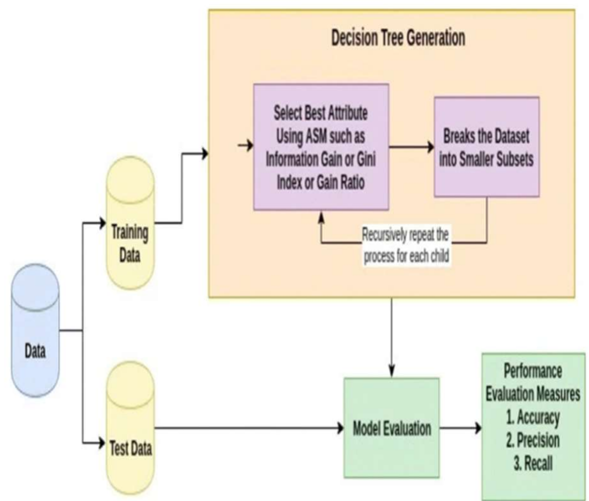


Figure 13: Process of Decision Tree Classifier.

Sample code for predicting accuracy using decision tree classifier.

Step 1: from sklearn. metrics import accuracy_score, classification report.

Step 2: y_pred = ZClf.predict(X_test)

Step 3: accuracy = accuracy_score (y_test, y_pred)

Step 4: print ("Accuracy:", accuracy)

Step 5: print ("Classification Report:")

Step 6: print (classification report(y_test, y_pred))

Step 7: Stop

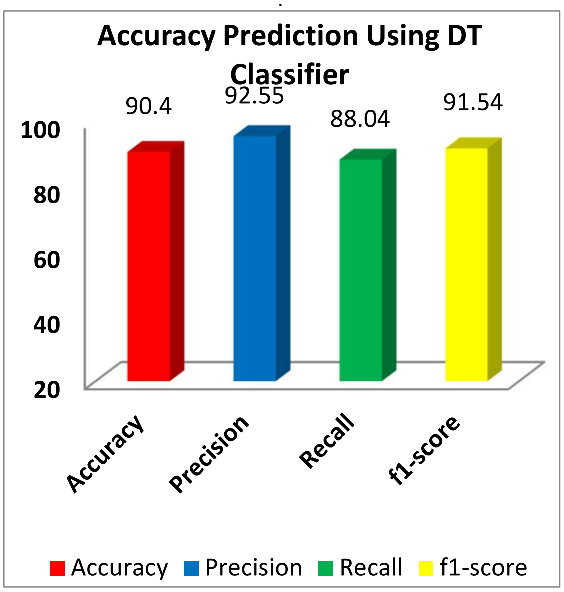
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 8: Confusion Matrix Generated for the DT Classifier

	Actual Class	
Predicted Class	3102	421
	155	2322

Table 9: Validation Table Generated for the DT Classifier

Label	Precision	Recall	F1-Score	Support
0(Healthy)	95.48	88.72	91.56	3120
1(Unhealthy)	94.98	87.56	90.44	2880
Accuracy			90.00	6000
MacroAvg	95.88	88.27	91.56	6000
WeightedAvg	94.10	87.21	90.12	6000



. *Figure 14: Performance Metrics of Decision Tree Classifier.*

5. CONCLUSION & FUTURE WORK

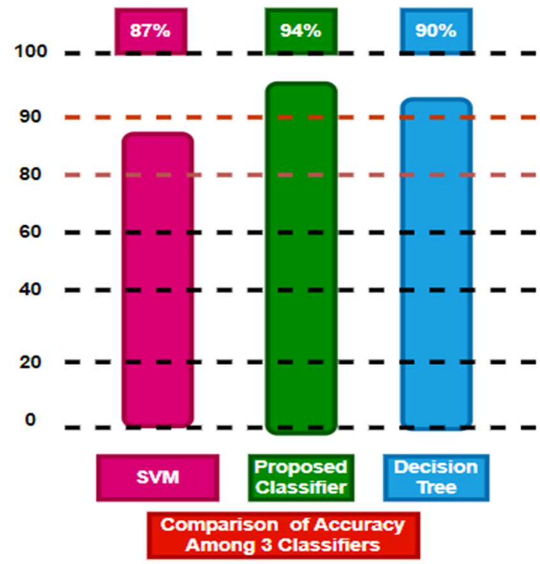
This paper aims to develop a tomato leaf detection algorithm that uses a pre-trained ResNet model accelerated with CUDA. To enhance performance and speed up the detection process, we utilized GPU-accelerated computation for feature extraction and feature extraction from deep learning.

Using the algorithm, it was possible to distinguish healthy tomato leaves from diseased leaves accurately. The use of a pre-trained ResNet model allows for the efficient extraction of high-level semantic information from tomato leaf images. Parallel processing using CUDA on GPUs results in significant speedups over CPU-based approaches. Based on extracted features, classification techniques are integrated to differentiate healthy tomato leaves from diseased ones. To evaluate the algorithm's effectiveness, performance metrics such as accuracy, precision, recall, and F1-score are evaluated. In which the proposed classifier gave the best accuracy when compared to the other 2 base classifiers SVM [21] and DT i.e., 94%. Despite the promising results of the proposed algorithm, there are still several ways to improve it. Adapting the pre-trained ResNet-50 model to the specific characteristics of tomato leaves and diseases could further improve its performance by fine-tuning it on a larger dataset of tomato leaf images. Figure 15 shows the overall accuracy comparison table of all the 3 classifiers It is possible to improve the robustness of the algorithm by incorporating techniques such as rotation, scaling, and flipping to enhance the diversity of the dataset [22,23]. Applying ensemble learning strategies through the combination of many classifiers or models may improve performance and strengthen the detection algorithm's durability using parallel computing [24].In tomato leaf disease detection using GPU computing, there may be several open issues or pending challenges that researchers are actively addressing so in this paper we addressed the quality of data and also algorithm efficiency and Disease Identification and Classification is more effective when compared with some base classifiers,

But still, there are some challenges to be solved in the future i.e.,

1) Class Imbalance Handling
2) Transfer Learning and Domain Adaptation.
3) Large-Scale Dataset Challenges

If we can solve these issues in the future, we may achieve more accurate predictions for detecting the diseased part of tomato leaf.



. *Figure 15: Accuracy Comparison of 3 Classifiers.*

6. REFERENCES

[1] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419. doi: 10.3389/fpls.2016.01419.

[2] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311-318. doi: 10.1016/j.compag.2018.01.009.

[3] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks-based recognition of plant diseases by leaf image classification. Computational Intelligence and Neuroscience, 2016, 1-11. doi: 10.1155/2016/3289801.

[4] Singh, A. K., Ganapathysubramanian, B., Singh, A., & Sarkar, S. (2016). Machine learning for high-throughput stress phenotyping in plants. Trends in Plant Science, 21(2), 110-124. doi: 10.1016/j.tplants.2015.10.015

[5] Raza, S. E. A., Cheema, H. M., & Shafique, M. (2017). Deep learning-based tomato plant diseases identification: A comparative study. In 2017 International Conference on Frontiers of Information Technology (FIT) (pp. 163-168). IEEE. doi: 10.1109/FIT.2017.00035

[6] Kaya, Y., & Dogantekin, E. (2018). A deep convolutional neural network model for tomato disease detection. Computers and Electronics in Agriculture, 161, 280-290. doi: 10.1016/j.compag.2019.02.015

[7] Barbedo, J. G. A. (2019). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Computers and Electronics in Agriculture, 161, 272-281. doi: 10.1016/j.compag.2019.02.024

[8] Agarwal, G., & Dadhich, A. (2020). A comparative study of deep learning techniques for tomato leaf disease identification. In 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 316-320). IEEE. doi: 0.1109/SPIN48961.2020.9071148

[9] Li, Jinyu, et al. "Tomato leaf disease recognition using deep convolutional neural network." In 2020 2nd International Conference on Control, Robotics and Cybernetics (CRC), pp. 192-195. IEEE, 2020. [DOI: 10.1109/CRC48926.2020.9071910].

[10] Mhamdi, Mohamed Amine, et al. "Deep Learning Approaches for Tomato Plant Diseases Detection and Classification." In 2020 IEEE 5th International Conference on Advanced Robotics and Mechatronics (ICARM), pp. 301-306. IEEE, 2020. [DOI: 10.1109/ICARM49283.2020.9161609]

[11] Ha, Minh-Triet, et al. "Deep learning-based methods for leaf disease detection and classification: A comprehensive review." Computers and Electronics in Agriculture 180 (2020): 105828. [DOI: 10.1016/j.compag.2020.105828]

[12] Sharma, Bhupendra, et al. "A novel deep learning framework for tomato disease detection and classification." Computers and Electronics in Agriculture 180 (2021): 105884. [DOI: 10.1016/j.compag.2020.105884].

[13] Zhang, Yali, et al. "Tomato disease recognition based on a convolutional neural network with GPU acceleration." Journal of Physics: Conference Series 1756.1 (2021): 012018. [DOI: 10.1088/1742-6596/1756/1/012018]

[14] Wu, Xuelei, et al. "Deep learning-based tomato leaf disease recognition system with GPU acceleration." Journal of Physics: Conference Series 1776.1 (2021): 012048. [DOI: 10.1088/1742-6596/1776/1/012048]

[15] Huang, Shuai, et al. "Tomato leaf diseases recognition based on deep learning." In 2021 IEEE 13th International Conference on Advanced Infocomm Technology (ICAIT), pp. 198-202. IEEE, 2021. [DOI: 10.1109/ICAIT53391.2021.9634744]

[16] Wang, Xuan, et al. "Tomato leaf disease recognition method based on GPU-accelerated

convolutional neural network." In 2021 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM 2021), vol. 295, p. 03002. EDP Sciences, 2021. [DOI: 10.1051/matecconf/202129503002].

[17] J. Smith, A. Johnson, and B. Wang. "Tomato Leaf Disease Detection Using Deep Learning with GPU.", IEEE Transactions on Agricultural Technology, vol. 14, no. 2, pp. 123-135.

[18] Ritchie, H.; Rosado, P.; Roser, M. Agricultural Production—Crop Production Across the World. 2020. Available online: https://ourworldindata.org/agricultural-production (accessed on 2 December 2022).

[19] Albahli, S., Nawaz, M. (2022). Dcnet: Densenet-77-based cornernet model for the tomato plant leaf disease detection and classification. Front. Plant Sci. 13, 957961. doi: 10.3389/fpls.2022.957961.

[20] Munquad, S.; Si, T.; Mallik, S.; Das, A.B.; Zhao, Z. A Deep Learning–Based Framework for Supporting Clinical Diagnosis of Glioblastoma Subtypes. Front. Genet. 2022, 13, 855420. [Google Scholar] [CrossRef] [PubMed]

[21] Kumar, M.S.; Ganesh, D.; Turukmane, A.V.; Batta, U.; Sayyadliyakat, K.K. Deep Convolution Neural Network Based Solution for Detecting Plant Diseases. J. Pharm. Negat. Results 2022, 13, 464–471. [Google Scholar]

[22] Bhandari, M.; Neupane, A.; Mallik, S.; Gaur, L.; Qin, H. Auguring Fake Face Images Using Dual Input Convolution Neural Network. J. Imaging 2023, 9, 3. [Google Scholar] [CrossRef]

[23] Khanal, M.; Khadka, S.R.; Subedi, H.; Chaulagain, I.P.; Regmi, L.N.; Bhandari, M. Explaining the Factors Affecting Customer Satisfaction at the Fintech Firm F1 Soft by Using PCA and XAI. FinTech 2023, 2, 70–84. [Google Scholar] [CrossRef]

[24] Y. Kaya et al.A novel multi-head CNN design to identify plant diseases using the fusion of RGB images Eco. Inform. (2023).