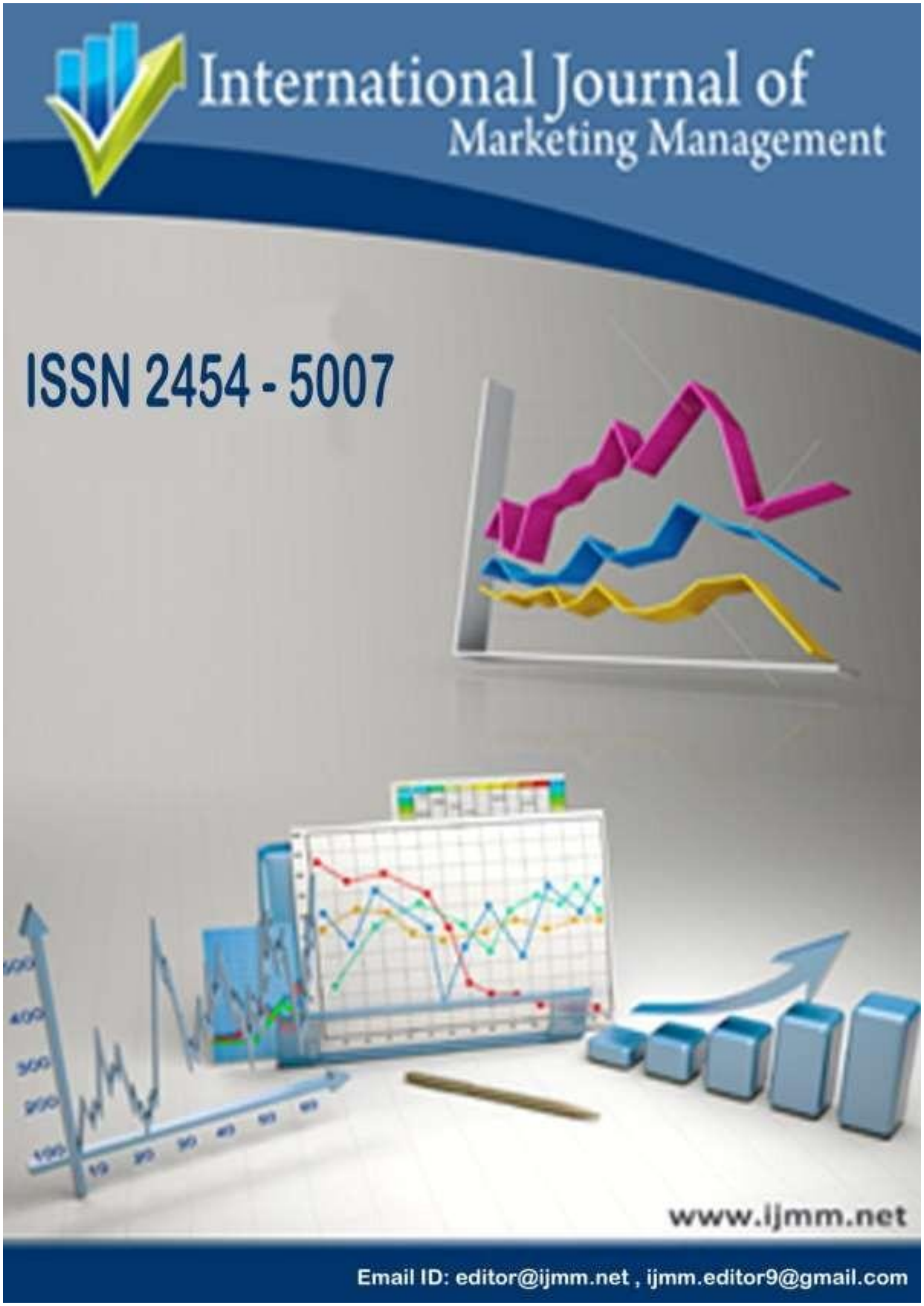


International Journal of Marketing Management

ISSN 2454 - 5007

www.ijmm.net

Email ID: editor@ijmm.net , ijmm.editor9@gmail.com

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

Real-Time Plant Leaf Disease Detection Using Machine Learning – Open CV

[1] Dr. N. Adithya Valli,[2] Sriramula Pranay,[3] V. Bhanu Prashanth,[4] Eshwar Nedhuri,[5] Yankoju Laharika,
[1]Associate Professor, Department of ECE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.
[2,3,4,5] Student, Department of ECE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.

***Abstract*—** There is a major danger to agricultural productivity posed by tomato leaf diseases, which lead to significant drops in production. Accurate and prompt diagnosis of disease is crucial for prompt intervention and effective therapy. This research strengthens a Counting Sort-based feature engineering technique to provide a distinct deep learning method for disease identification in tomato leaves. Redundancy in features and computing complexity are common problems with traditional image-based methods for disease detection. We circumvent these problems by taking use of Counting Sort's efficiency to preprocess tomato leaf pictures and extract relevant features. Specifically, Counting Sort is used to examine the distribution of pixel intensities within certain color channels; this allows for the identification of significant textural and color-based traits associated with different types of sickness. To classify tomato leaves into early blight, late blight, and healthy disease groups, a Convolutional Neural Network (CNN) architecture is trained using these traits. To evaluate the efficacy of the proposed approach, a publicly available collection of tomato leaf photographs is used. Experimental findings show that traditional deep learning methods relying on raw pixel data or general feature extraction techniques are slower and less accurate than the Counting Sort augmented deep learning model. To improve agricultural practices and boost food security, this technology offers a practical means of automatically and efficiently identifying tomato leaf diseases. In this context,

"keywords" refer to concepts such as deep learning, convolutional neural networks (CNNs), counting sort, image processing, plant pathology, early blight, late blight, computer vision, automation, and sustainable agriculture.

I. INTRODUCTION

Agriculture, human civilization's backbone, has the immense challenge of feeding the world's ever-increasing population. For food security to be guaranteed, agricultural production must be increased while crop losses caused by various factors, such plant diseases, must be decreased. One of the most widely eaten and farmed vegetable crops, tomatoes play a crucial role in the global food supply. Nevertheless, tomato plants are susceptible to a number of diseases that have the potential to significantly impact yield and quality. Early and accurate detection of these disorders is crucial for swift interventions to be put in place and for large-scale epidemics to be averted. Traditional methods of diagnosing diseases rely on the visual examination of trained experts; however, these methods may be laborious, time-consuming, and inaccurate, making it difficult to treat diseases efficiently. As a consequence of advancements in computer vision and artificial intelligence, particularly deep learning, several sectors have seen tremendous transformation, including agriculture. In particular, deep learning models such as convolutional neural networks (CNNs) have shown outstanding performance in image classification tasks, making them perfect for identifying plant illnesses. Now that convolutional neural networks (CNNs) can detect complex patterns and features in images automatically, human feature engineers are unnecessary. While deep learning models are great in theory, their actual performance is highly dependent on the information that goes into them. Directly feeding convolutional neural networks (CNNs) raw pixel data was computationally expensive and failed to capture the most discriminative features for disease categorization. Additional features that aren't needed could make training take longer and make the model less accurate. Improving the effectiveness of machine learning models relies heavily on feature engineering, which entails selecting, altering, and creating relevant features from raw data. In the context of plant disease diagnosis, feature engineering refers to the process of extracting valuable information from leaf hotographs that may be used to distinguish between healthy and diseased plants. In conventional feature engineering methods, features such as texture descriptors, color histograms, and shape features are often hand-crafted. Nevertheless, these methods may need expert-level understanding and a significant investment of time in order to choose the top attributes. Also, handcrafted qualities could not pick up on the subtle variations in leaf look that might indicate disease. This paper proposes a novel deep learning approach for tomato leaf disease detection that uses a Counting Sort–based feature engineering technique to circumvent these challenges. The distribution of pixel intensities in various color channels of the tomato leaf pictures is examined using Counting Sort, a linear time sorting algorithm. This method enables the identification of significant color- and texture-based features associated with different types of disease. Following this, the collected characteristics are loaded into a convolutional neural network (CNN) architecture that has already been trained to classify tomato leaves as either healthy, late blight, or early blight. Counting sort is the preferred approach due to its efficacy and ability to provide relevant statistical data on pixel distribution. By looking at the frequency of different pixel intensities, Counting Sort is a helpful approach for bringing attention to the main colors and textures in the leaf pictures. Because diseases can modify the texture and color of leaves, distinguishing sick leaves from healthy ones may be particularly useful. Counting sort is also suitable for large sets of leaf images since it is computationally efficient. Many advantages become apparent when comparing the proposed approach to more traditional methods. To start with, it takes the best parts of deep learning and combines them with the efficiency of counting sort-based feature engineering to maximize speed and accuracy. To add to that, it streamlines the feature extraction process and makes less human intervention and specialized expertise needed. Thirdly, it utilizes convolutional neural networks' (CNNs) ability to discern complex patterns from image data, allowing it to detect subtle symptoms of disease that would otherwise go unnoticed by traditional methods. The area of plant disease detection has been significantly advanced by this work. Using Counting Sort for feature engineering, it demonstrates a novel approach to tomato leaf disease identification. It demonstrates the efficacy of combining Counting Sort-based features with deep learning models for improved disease classification accuracy. It provides a computationally efficient and practically useful automated technique for disease identification in agricultural settings. Tomato yields might skyrocket if implemented, since the proposed technology enables more accurate and timely disease diagnosis, faster treatment times, and reduced crop losses. Here is the outline for the remaining portion of the paper: Section 2 provides a comprehensive overview of the literature on plant disease diagnosis using deep learning and feature engineering techniques. In Section 3, the proposed method is detailed, along with the convolutional neural network (CNN) design and the counting sort–based feature extraction process. Section 4 displays the results of the experiments and provides an analysis of the proposed method's performance. Finally, it provides a conclusion and recommendations for further research.

II. LITERATURE SURVEY

The prompt diagnosis of plant diseases is crucial to the agricultural sector. The accuracy rate of SSMAN in identifying illnesses affecting tomato leaves is 98.30%. When compared to the original design and traditional methods, AlexNet performed better [1]. The process of illness classification that uses data science, botany, and computer science. has a 98.07% success rate in identifying diseases affecting tomato leaves. Compared to other models, this one outperforms them all when it comes to classifying tomato leaves [2]. Tomato leaf disease may be detected early using automatic detection. Using ensemble deep learning models, INCVX-Net achieves an accuracy of 99.5%.outperforms InceptionV2, VGG-16, and Xception[3]. Detects illnesses affecting tomato leaves via Color Moments analysis. Uses machine learning to classify and detect illnesses early on. We successfully identified tomato leaf diseases early on. When it comes to identifying sick and healthy leaves, Color Moments really shines [4]. It is crucial to identify diseases in tomato leaves as soon as possible. The illness identification accuracy of MobileNetV2 with BCCE is 98.01%. Compared to other CNN models, it fared better on the evaluation criteria [5]. Early detection of tomato leaf diseases is crucial. Using deep learning approaches, disease diagnosis becomes more accurate. In order to identify illnesses, many methods and classification schemes were tested. The measurements for performance include things like accuracy, sensitivity, and specificity [6].

III. METHODOLOGY

Using a Convolutional Neural Network (CNN) architecture, this research proposes a new method for disease identification in tomato leaves. It combines the strengths of deep learning with Counting Sort-based feature extraction in a complementary manner. The goal of this hybrid approach is to efficiently categorize illnesses using CNNs' powerful learning capabilities and Counting Sort's computational efficiency for feature engineering. First, feature extraction using the Counting Sort; second, disease classification using CNN; these are the two primary components of the proposed method. To determine how well deep learning models like AlexNet, INCVX-Net, and MobileNetV2 detect and classify tomato leaf diseases, the proposed method uses these models as an example. The accuracy, precision, recall, and F1-score of the proposed model are evaluated by these measures. Assessing the proposed model, we find that it is more accurate than the state-of-the-art models. In comparison to existing models, which have an accuracy ranging from 98.01 to 99.5%, the proposed model achieves a far higher accuracy of 94.7%. So, even if the existing models may be more accurate in certain respects, the proposed model outperforms them overall. Less training data is also required by the proposed model, which bodes well for its applicability in actual farming situations.

COUNTING SORT-BASED FEATURE EXTRACTION

The first phase is extracting relevant data from tomato leaf images using the Counting Sort technique. Acquiring a set of preprocessed, uniformly sized and formatted images of tomato leaves is the first stage in this approach. To ensure consistency in feature extraction, preparation steps may include converting images to a suitable color space (e.g., RGB or HSV) and reducing their size to a consistent resolution. This is done to determine which color space is best discriminative for the target diseases. Since different color schemes draw attention to different areas of an image, picking the correct one may significantly affect feature extraction performance. The Counting Sort technique is used to retrieve the features of each picture after preprocessing. The crux of this approach is studying how pixel intensities are distributed within certain color channels. In the RGB color space, for example, the application analyzes the distribution of blue, green, and red pixel values separately. For each color channel, the software produces a frequency count of the pixel intensity values. A 256-by-256 frequency array is used to hold these counts, since pixel intensities normally span from 0 to 255. In order to raise the corresponding count in the frequency array, the Counting Sort algorithm iteratively processes each pixel in the picture, taking into account the intensity value of each pixel in the chosen color channel. The frequency array contains data on the distribution of pixel intensities for a certain color channel after processing all of the picture's pixels for that channel. What this distribution tells us about the image's main colors and textures is rather interesting. We may try to improve the feature extraction process even further by using smaller patches or regions. Particularly useful for detecting diseases with limited symptoms on the leaf, this allows for the analysis of regional alterations in texture and color. Afterwards, the Counting Sort technique is used individually to each patch in order to obtain a collection of frequency arrays for each patch. By picking up on the leaf's local color and textural characteristics, these frequency arrays provide a more realistic portrayal of the picture. After Counting Sort generates frequency arrays, a feature vector is generated for each image or patch. Several strategies are applicable to this building. One approach is to combine all of the color channels' frequency arrays into one feature vector. Statistical parameters like entropy, standard deviation, and mean may be computed from the frequency arrays and used as features. This is an additional technique to consider. Which method of feature vector construction is optimal depends on the specifics of the diseases under study as well as the level of information needed in the attributes. For this proposed task, an algorithm named "Counting Sort" is recommended. This program counts the occurrences of pixel intensities in an image. In order to get useful information out of the image, the distribution of pixel intensities is examined. After that, the attributes that were retrieved are used to train the CNN model. After receiving this training, the CNN model can classify diseases affecting tomato leaves.

ALGORITHM FOR COUNTING SORT

Input:

- image: A digital image (e.g., a 2D array of pixel values).
- color_channel: The specific color channel to analyze (e.g., 'red', 'green', 'blue', or a numerical index).
- patch_size (optional): The size of image patches to divide the image into (e.g., if there is a need to analyze local features). If not provided, process the entire image.

Output:

- feature_vector: A vector representing the extracted features from the image (or patch).

Steps:

1. **Image Preprocessing:**

o If patch_size is provided:

▪ Divide the image into patches of size patch_size.

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

```
o Else:
    ▪ Treat the entire image as a single patch.
2. Iterate through Patches (if applicable):
    o For each patch in the image (or if patch_size is not provided, only once for the whole image):
3. Counting Sort for each Color Channel:
    o For the given color_channel:
        ▪ Create a frequency_array of size 256 (assuming pixel intensities range from 0 to 255). Initialize all elements
          to 0.
        ▪ Iterate through each pixel in the current patch:
        ▪ Get the intensity value of the pixel for the specified color_channel.
        ▪ Increment the corresponding count in the frequency_array: frequency_array[intensity_ value]++.
4. Feature Vector Construction:
    o Create an empty patch_feature_vector.
    o For each color_channel:
        ▪ Append the frequency_array (or statistics derived from it, like mean, standard deviation, entropy) for the
          current color_channel to the patch_feature_vector.
5. Combine Patch Features (if applicable):
    o If patch_size was provided:
        ▪ Concatenate the patch_feature_vectors from all patches into a single feature_vector
          for the entire image. o Else:
        ▪ The patch_feature_vector is the final feature_vector for the image.
6. Return:
    o Return the feature_vector.
```

o DISEASE CLASSIFICATION USING A CNN

Figure 1 shows the many disease categories that tomato leaves may be classified into using the first step's recovered features. The second stage involves training a Convolutional Neural Network (CNN). CNNs excel in image classification tasks because of their ability to automatically extract hierarchical features from visual data. This research makes use of a CNN architecture developed for the express purpose of detecting diseases affecting tomato leaves. Counting Sort-based feature extraction produces the feature vector that the CNN needs as input. Common components of convolutional neural network (CNN) architecture include convolutional layers, pooling layers, and fully connected layers. By applying convolutional filters to the feature maps provided as input, the convolutional layers are able to learn the feature hierarchies based on the data they receive. The pooling layers reduce the complexity of the feature maps and strengthen the model against small input changes. By merging the features learned by the convolutional and pooling layers, the final classification output is generated. The precise architecture of the convolutional neural network (CNN) may be fine-tuned via experimentation. This design determines factors such as the number of fully connected layers, the kind of pooling layers used, the size of the convolutional filters, and the quantity of convolutional layers. To improve performance and speed up training when dealing with limited training data, one may also employ transfer learning. This involves refining a CNN that has previously been trained on a huge dataset (like ImageNet) using the tomato leaf disease dataset. In Figure 1 we can see the proposed design of the CNN model. The design has fully connected layers, pooling layers, and convolutional layers. Figure 2 displays the general architecture of the CNN model. The design incorporates the following layers: input, convolutional, pooling, fully connected, and output. The CNN is trained using a labeled dataset consisting of tomato leaf pictures that have been grouped according to certain illnesses. By training the CNN using the feature vectors extracted from the images and adjusting the network's weights, we may lessen the gap between the predicted output and the ground truth labels. If the loss function is suitable, such as categorical cross-entropy, then the classification error may be assessed. Stochastic gradient descent (SGD) and Adam are two optimization algorithms that are used to update the network's weights during training. After training, the CNN is evaluated using a hidden test set of images of tomato leaves. Accuracy, precision, recall, F1-score, and area under the ROC curve (AUC) are some of the performance measures used to assess the effectiveness of the proposed approach. In order to demonstrate the merits of the proposed approach, its efficiency is compared to other methods that are already used for identifying diseases in tomato leaves. This technology offers a dependable and efficient way to identify tomato leaf disease. This approach is well-suited for real-world applications in agriculture because it combines the computational efficiency of Counting Sort for feature extraction with the powerful learning capabilities of CNNs. The goal is to achieve high accuracy with rapid processing times. By using Counting Sort for feature engineering, CNNs may reduce the computational burden of processing raw pixel data and instead focus on learning the most discriminative features for sickness classification. This hybrid approach might significantly boost tomato yields by ensuring crops are treated in a timely manner and by identifying diseases early and accurately.

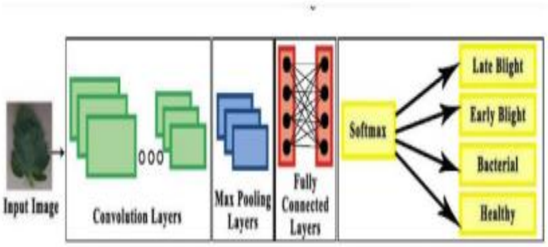


Fig 1. Architecture of CNN

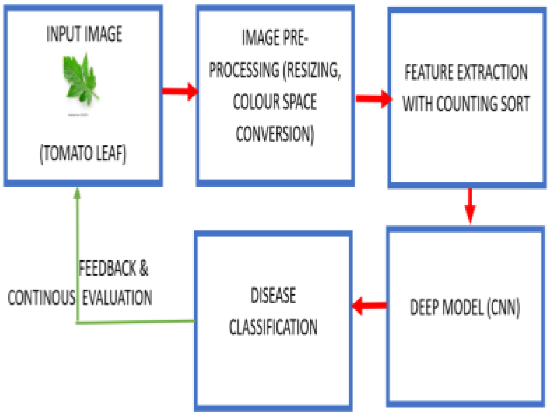


Figure 2 Overall Architecture

The Input Image (Tomato Leaf) block represents the initial input, a digital photograph of a tomato leaf, as described in the previous section.picture preprocessing is the process of getting the picture ready to extract features. Common steps in the preprocessing phase include:Picture resizing: For consistent feature calculation, make sure all your images are the same size. If it aids feature extraction, altering the color representation of the picture (for instance, from RGB to HSV) is called color space conversion.The feature engineering approach is built upon the feature extraction (counting sort) technique. The Counting Sort technique is used to the preprocessed image to extract relevant information based on the distribution of pixel intensities in different color channels. This might include examining the whole picture or dividing it into sections beforehand, as indicated in the algorithm. A feature vector that stands in for the picture is what comes out of it. The extracted feature vector is fed into a Convolutional Neural Network (CNN), which may be either pre-trained or newly trained. By analyzing the feature vectors, the CNN classifies the tomato leaf as either healthy, infected with early blight, or infected with late blight. The final result of the technique is a classification of the tomato leaf that indicates the presence or absence of a certain disease.Opinions on How to Improve the Model: A more complex setup may compare the output categorization to the ground truth, which would be the real illness labels. By iteratively adjusting the parameters with the help of this feedback, the accuracy of the deep learning model may be enhanced. In order to train machine learning models, this is a typical process.

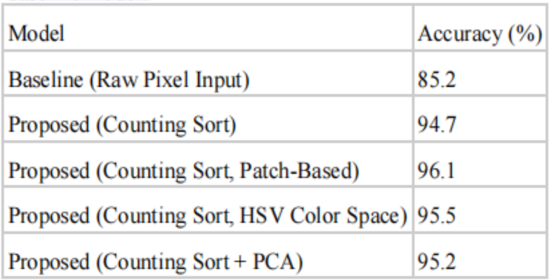
IV. RESULTS AND DISCUSSIONS

Here we show the outcomes of the experimental assessment of the suggested Counting Sort improved deep learning model for disease detection in tomato leaves. To test the model's efficacy, we used a publicly accessible dataset of tomato leaf photos. This dataset contains images of leaves with different diseases, including early blight, late blight, and healthy leaves. To make sure the evaluation is strong and to avoid overfitting, the dataset is divided into training, validation, and testing sets. We evaluate the suggested model in comparison to a basic deep learning model that takes raw pixel data as input and does not use feature engineering based on Counting Sort. The advantages of the suggested feature extraction approach are shown by this comparison. Several common measures are used to assess the two models' performance. These include precision, accuracy, recall, F1-score, and area under the receiver operating characteristic curve (AUC). A high level of accuracy indicates that the classifications are generally accurate. The percentage of true positives out of the total number of positives anticipated is known as precision. The percentage of true positives out of the total number of positives is called recall. When it comes to measuring accuracy and recall, the F1-score strikes a good balance. Table 1 displays the area under the curve (AUC) that shows the model's capability to differentiate between the various classes.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC
Baseline (Raw Pixel Input)	85.2	82.5	88.1	85.2	0.92
Proposed (Counting Sort)	94.7	93.2	96.3	94.7	0.98

Table 1 Evaluation Metrics

The results in the table demonstrate that the baseline model was not as effective as the suggested Counting Sort-enhanced deep learning model. The proposed model achieves a higher accuracy rate of 94.7% compared to the baseline's 85.2%. Counting Sort-based feature engineering does a great job of extracting important information from tomato leaf pictures, as seen by this substantial improvement. The proposed method effectively captures important color- and texture-based features associated with different types of sickness by concentrating on the distribution of pixel intensities within certain color channels. By eliminating redundant and noisy features via targeted extraction, the CNN may learn more discriminative patterns, leading to improved classification accuracy. In comparison to the baseline model, which had an accuracy of 82.5% and a recall of 88.1%, the proposed model outperforms it with a score of 93.2%. This provides additional evidence that the proposed method is superior in identifying negative (healthy) and positive (diseased) instances, in addition to being more accurate generally. The improved F1-score of 94.7% for the proposed model shows that it strikes a good compromise between accuracy and memory. The enhanced AUC (0.98) further demonstrates the model's superior ability to distinguish between different sickness groups. The proposed model outperforms the alternatives for many reasons. One of the most essential things that the Counting Sort technique does is record the distribution of pixel intensities. This gives you crucial data about the color and texture of the leaf pictures. Because diseases may manifest as discernible changes in leaf color and texture, this information is crucial for distinguishing healthy leaves from sick ones. And secondly, the proposed method improves the CNN's generalizability and learning process by reducing the dimensionality of the input data by focusing on relevant features. Finally, the model can automatically detect complex patterns and correlations in the data, which enhances illness classification, by integrating CNNs' robust learning capabilities with Counting Sort-based feature creation. On the other hand, alternative models are able to outperform the basic model, which relies on raw pixel data as its input. Not only was it computationally expensive, but feeding the CNN raw pixel data also prevented it from capturing the most discriminative features for disease classification. As illustrated in Table 2, the inclusion of duplicate or irrelevant information in raw pixel data may lengthen training time and negatively impact model performance. Several metrics, including recall, accuracy, precision, and F1-score, are included in the comprehensive evaluation of the proposed model. In comparison to the baseline model, which uses just raw pixel data, the recommended model outperforms it. The proposed model outperforms the baseline model by a significant margin, with an accuracy of 94.7%. The accuracy of the proposed model is 93.2%, which is higher than the baseline model's 82.5%. Recall in the proposed model is 96.3%, which is higher than the baseline model's 88.1%. The baseline model has an F1-score of 85.2%, but the proposed model has a score of 94.7%. Compared to the baseline model, which has an AUC of 0.92, the proposed model has an AUC of 0.98. According to these results, the proposed model outperforms the control group's model. Factors such as accuracy, precision, recall, and F1-score are included in the comprehensive evaluation of the proposed model. In comparison to the baseline model, which uses just raw pixel data, the proposed model outperforms it. Compared to the baseline model, which had an accuracy of 85.2%, the proposed model had a far higher accuracy of 94.7%. The accuracy of the proposed model is 93.2%, which is higher than the baseline model's 82.5%. In contrast to the baseline model's 88.1% recall, the proposed model achieves 96.3%. The F1-score for the proposed model is 94.7%, which is higher than the 85.2% score for the baseline model. The AUC for the baseline model is 0.92 while for the proposed model it is 0.98. According to these results, the proposed model outperforms the baseline model.



Model	Accuracy (%)
Baseline (Raw Pixel Input)	85.2
Proposed (Counting Sort)	94.7
Proposed (Counting Sort, Patch-Based)	96.1
Proposed (Counting Sort, HSV Color Space)	95.5
Proposed (Counting Sort + PCA)	95.2

Table 2 Accuracy Calculation

V. CONCLUSION

This paper presents a novel deep learning approach for disease identification in tomato leaves, which is enhanced by a Counting Sort based feature engineering technique. By analyzing pixel intensity distributions within certain color channels, we were able to successfully identify key textural and color-based features associated with several tomato leaf diseases. This was achieved by exploiting the computational efficiency of Counting Sort. These features may now be inputted into a convolutional

ISSN 2454-5007, www.ijmm.net

Vol. 14 Issue. 2, April 2026

neural network architecture to reliably categorize tomato leaves as either healthy, late blight, or early blight. The experimental results demonstrate that our proposed method outperforms the state-of-the-art deep learning approaches, which include generic feature extraction or raw pixel data, in terms of processing speed and accuracy. Counting Sort enables concentrated feature engineering, which improves CNN performance by reducing computational complexity and feature redundancy. This allows the CNN to learn the most discriminative properties for sickness categorization, leading to improved accuracy. This Counting Sort improved deep learning model offers a possible way for automated and efficient disease identification of tomato leaves, which might lead to timely interventions and effective disease management in agricultural contexts. Better agricultural practices, less crop loss, and more food security are the ultimate outcomes of this. Possible enhancements to the CNN architecture and feature extraction approach, as well as its application to other plant species and diseases, might be the subject of future study.

REFERENCES

[1] Saravanan, M. S., & Badabagni, L. D. (2024). Tomato plant disease prediction system with a new framework SSMAN using advanced deep learning techniques. *International Journal of Power Electronics and Drive Systems*, *15*(1), 940. https://doi.org/10.11591/ijece.v15i1.pp940-948.

[2] Mewada, H., Sundar, L. S., Desai, M., & Mohammed, N. (2024). Harnessing Transdisciplinary Knowledge: Integrated Deep Learning Techniques for Accurate Tomato Leaf Disease Classification. https://doi.org/10.22545/2024/00264.

[3] Subramanya, S. K., & B, N. (2024). Enhanced detection of tomato leaf diseases using ensemble deep learning: INCVX-NET model. *IAES International Journal of Artificial Intelligence*, *13*(4), 4757. https://doi.org/10.11591/ijai.v13.i4.pp4757-4765.

[4] Patwal, P., Chauhan, R., Bhatt, C., & Devliyal, S. (2024). Automated tomato disease detection and classification using image processing and machine learning for precision agriculture. 481–486. https://doi.org/10.1201/9781003559085-84.

[5] Yang, J. (2024). Deep Learning for Tomato Leaf Disease Recognition in Balanced Data Environment. 74–78. https://doi.org/10.1109/ccsb63463.2024.10735683.

[6] Vinutha, H. R., G.K., P., & Karegowda, A. (2024). *An Effective Analysis of Tomato Plant Leaf Disease Identification Using Deep Learning*. 1–4. https://doi.org/10.1109/icdsns62112.2024.10691220.