



ISSN: 0067-2904

Quality Assessment of South Indian Cherry Tomatoes Based on Deep and Convolutional Neural Networks

K. Venkatesh Sharma¹, Mahammad Mastan², G. Jai Arul Jose^{3*}, Louay A. Hussein Al-Nuaimy²

¹ CVR college of Engineering

² Oman College of Management & Technology

³ BMS Institute of Technology and Management, India

Received: 6/2/2023

Accepted: 25/8/2023

Published: 30/10/2024

Abstract

A few decades ago, quality assessment was a significant challenge in agriculture, even with tiny amounts of agricultural products. They are subject to diseases, pesticides, and environmental variables that have led to challenges to quality and safety. The quality evaluation of these items serves a crucial function in quality control and increases productivity before consumption. This paper proposes a deep neural network for solving Indian tomato diversity problems based on the assessment of their features. This study was carried out using four different categories of south Indian cherry tomatoes, i.e., spot, BER, calyx, and non-calyx. Thirty (30) relevant features were taken from an RGB-HIS color space: textural color moments, gray level co-occurrence matrix (GLCM), and defective proportion. They were then normalized in MS-EXCEL with a Min-Max function and fed into a multilayer perception deep neural network (MLP-DNN) and a convolutional neural network (CNN), respectively. The MLP-DNN was made with ten (10) hidden layers and was used with a fine-tuned convolutional neural network (CNN) of Alex net architecture with eight (8) learned layers to do grading and detection separately using MATLAB 2018a software. The results of training, testing, and validating both networks showed that they did an excellent job of solving the two-class problem (good and bad), with an overall prediction accuracy of more than 97% and a bit of loss function in both cases. However, the algorithms can be improved and employed to detect defects and sort fruits or vegetables of all kinds in the agro-industry with a larger dataset to reduce overfitting and increase validation accuracy in the future.

Keywords: Quality Assessment, Deep Neural Network, Convolutional Neural Network, South Indian Cherry Tomato, MLP-DNN.

1. Introduction

Agriculture is a well-known industry that enables people to farm, plant, and harvest crops. However, technical advancements have resulted in a growing amount of agricultural food that meets present needs while also providing a reserve for the future. The foods and beverages on which these products are based are subject to pathogen, insect, disease, and environmental concerns, which can negatively influence their insides and outsides [1].

Food safety has become a significant problem in many countries. Over 1.4 million outbreaks of various diseases have been connected to substandard food, making it an essential preventive measure in crop husbandry systems. According to one study, up to 50% of crop output losses could be attributed to exaggerated reasons. Tomatoes are one of the most

*Email: jaiaruljose@gmail.com

sought-after vegetables due to their plethora of uses. Even in the present era, its freshness, quality, and nutritional content remain critical. *Solanum lycopersicum* is grown worldwide due to its diverse usage in fresh and processed forms (ketchup, paste, and powder). India is the world's second-largest tomato grower, behind China, accounting for 11% of worldwide tomato production. Most tomatoes used in the country are fresh. At the same time, the remainder is prepared from tomato paste, tomato juice, tomato sauce, and ketchup tomatoes, among others. The fast-food industry has grown tremendously as disposable incomes have increased and food consumption preferences have become more westernized. Tomato-processing products, such as ketchup, are frequently paired with burgers, sandwiches, pizza, and fries. Additionally, tomato-based processed items such as tomato paste and tomato sauce are commonly utilized in Indian cuisines. Additionally, conveniences, a longer shelf life, urbanization, shifting food preferences, and the expansion of the organized retail sector in India are driving forces behind the tomato processing industry [2].

Cherry tomatoes, a popular ingredient in salads, curries, and sauces, particularly at star hotels and social gatherings, make their way to supermarket shelves and fruit and vegetable stands across India. Tomatoes are a significant crop in terms of economic value generated, with demand increasing rapidly due to their diverse consumption and health benefits [3]. Healthline.com classifies tomatoes as Roma, pear, cherry, beefsteak, heirloom, and grape. However, the United States Department of Agriculture (USDA) has identified and used a color chart to denote the six distinct stages of tomato maturity and ripeness for ease of sorting and classification. As a result, these stages of tomato fruit classification adhere to this order: green, breaker, transformation, pink, light red, and red [4]. Their ripening phases have an effect on their chemical and physiological quality due to the numerous probable flaws: early and late blight, radial and concentric cracks, bacterial spot/speck, anthracnose, blossom end rot, and calyx [5] [6] [7].

The outer look of these fruits retains consumer-observable faults, eroding their point-of-sale value and resulting in significant economic losses and health and safety concerns. However, tomato qualities such as texture, shape, size, color, shelf life, weight, defect-free status, and disease-free status influence their appeal to customers and serve as a barometer for their acceptance or rejection in the market when determining their quality [4][8][9].

Recent investigations have revealed that numerous researchers have utilized various algorithms for sorting and grading tomato quality. Still, each method had distinct drawbacks, including limited feature consideration, high-cost implications, and classification inaccuracy. However, research has been mostly focused on tomato classification based on the ripening stage, with limited information on application defect detection. As a result, this research presents a technique for assessing the quality of South Indian cherry tomatoes using supervised machine learning (DNN and CNN). The empirical datasets for the investigation were collected from Changchun Local Market in Jilin Province. The algorithms used were pre-processing, feature extraction and learning, segmentation, and classification. The DNN and CNN were chosen for their high speed and accuracy to fix the algorithms' flaws to the fullest extent possible while also accomplishing the following objectives:

1. Efficacy and constancy.
2. It requires less computational time.
3. Make the classifier more accurate.
4. Substantial reductions in the cost of labor.
5. Predictable for better quality.

2. Previous Studies

Mohit Agarwal et al. [6] proposed a simple CNN model with 8 hidden layers. Using the publicly available PlantVillage dataset, the proposed lightweight model outperforms traditional machine learning approaches and pre-trained models with 98.4% accuracy. The PlantVillage dataset contains 39 classifications of different crops, 10 of which are tomato illnesses. In pre-trained models, VGG16 outperforms k-NN with 93.5 percent accuracy. After picture augmentation, image pre-processing was utilized to improve the suggested CNN's performance. The presented model also works remarkably well on non-PlantVillage datasets, with 98.7% accuracy.

M.S. Iraj [10] recommended using artificial intelligence to forecast tomato quality classes. A multilayer architecture of a SUB-adaptive neuro-fuzzy inference system (MLA-ANFIS) was built using a tomato picture data set with seven input features from a farm. Instead of analyzing characteristics retrieved from tomato photos, a deep-stacked sparse auto-encoder (DSSAE) technique was proposed. The DSSAEs method was more accurate than prior methods and used a novel mechanism for grading tomato quality. The proposed design attained 83.2 percent sensitivity, 96.5 percent specificity, 89.40 percent g-mean, and 95.5 percent accuracy. It may therefore improve tomato inspection and processing quality.

P.Wan et al. [11] proposed a computer vision approach to detect tomato maturity. They used Visual C++ 6.0 to preprocess two Roma and pear tomato types with green, orange, and red ripening species. Following the Otsu threshold procedure, other morphological techniques were used to acquire the desired region. The resulting RGB values were then converted to the HSI color model and divided into five concentric circles. Their radii were marked to produce the hue averages of the five concentric color features. Then a backpropagation neural network (BPNN) was built to classify them. With a standard deviation of 1.2 percent, the research revealed an average accuracy of 99.31% for the three tomato ripening stages (green, orange, and red).

L. C. Santosh et al. [12] used an innovative approach for varietal discrimination and identifying Nepalese tomato cultivars using two sets of samples organized into a database blob. They calculated the shape, area, length, width, roundness, and color intensities. They then employed nCDA, pairwise nCDA-MSI, and PCA on the experimental data. Then PLS-DA was used to classify all the cultivars. These models have a discriminating accuracy of 100%, 85%, 80%, and 96%, respectively, with some sensitivity cultivars and cross-validation and prediction errors of 7%, respectively.

G. Liu et al. [13] suggested a Python 3.5 technique for intelligent tomato detection based on field climate parameters. Partitioned, intersected, and blocked RGB samples were taken, scaled by interpolation, and morphologically pre-processed. Scanners, false color removal (FCR), and non-maximum suppression (NMS) were used consecutively to identify samples with observed cases. They then used a histogram of an oriented gradient to overcome robot harvesting problems and trained the network with a support vector machine technique. Other framework procedures were used to analyze the proposed techniques' performance, and the algorithm's experimental findings were highly successful. N.El-Bendary et al. [14] used a multi-class support vector machine to classify tomato ripeness. They worked with five datasets containing 230 samples gathered according to USDA ripening phases and converted to HSV color space. They created their algorithms for pre-processing, feature extraction, and classification. Background subtraction was used to complete the pre-processing step. The characteristics were extracted using PCA and classified using SVM algorithms with three

different kernels (linear, RBF, and MLP). They achieved a classification accuracy of 92.72% using a linear kernel function.

However, spectral approaches are time-consuming, difficult to extract characteristics from, and require expensive gear like spectrometers, fiber optics, spheres, and lenses for high-resolution imaging. In contrast, an adaptive neurofuzzy system relies on subjective statistical and linguistic information, making it unreliable. Thus, there is a need to present machine vision and image processing algorithms that are more efficient, dependable, resilient, and cost-effective; therefore, a supervised learning neural network is proposed to analyze the quality of South Indian cherry tomatoes in real-time (DNN and CNN). A local market dataset from Changchun was used to test pre-processing, feature extraction/learning, segmentation, and classification using DNN and CNN.

3. Methodology (Methods and Materials)

The research work undertaken is segmented into four phases, which are listed and detailed as follows:

1. Data Collection Techniques
2. Experimental Setup Analysis
3. Proposed Framework Model
4. Algorithm

3.1 Data Collection Techniques

In this research, four different cherry tomatoes, as given in Figure 1, were obtained from the local market in Mehdipatnam, Hyderabad, Telangana, India. First, the samples were collected, washed, and cleaned, and then visually classified by an expert and later categorized into training and testing datasets. While the training set was utilized to build the recognition model, the test set was also used to verify the accuracy of the developed model. The breakdown analysis for the four kinds of samples is shown in Table 1 and Figure 1.

Table 1: Breakdown of Datasets (as collected from the Samples)

No	Defects	Red	Light Red	Total
1	Spot	90	37	127
2	Calyx	85	46	131
3	BER	75	38	113
4	No Calyx	95	34	129
Total Number of Acquired Cherry Tomatoes				500



Figure 1: Dataset Illustration (These samples were obtained from the local market in Mehdipatnam, Hyderabad, Telangana, India, and from [12])

3.2 Experimental Setup Analysis

It is a practical procedure that provides a layout of materials and tools. It entails arranging the hardware components in the proper order for the analysis to succeed. The experimental setup used consists of five critical hardware components: a lighting system, a PC, a

background (gabbler), a camera, and a sample (tomato). As illustrated and described in Figure 2, these components require an image acquisition system for quality evaluation. To cut down on classification and grading problems, labor costs, and time spent, an effective experimental setup was made with the above materials and tools, using a healthy and damaged South Indian tomato blossom end root (BER), a spot, and a healthy tomato with and without a calyx, based on their internal and external quality.

1. *Lighting system:* This includes the design, layout, and construction of the lighting unit, which controls the illumination component at the point of capture. Various lighting systems, such as incandescent, fluorescent, and so on, have been used. However, for this experiment, a three-set of an 18W machine vision light with separate color modes of red, green, and blue (RGB) was purchased and used, along with a single control unit. They have permanently installed it 65cm away from the background level, where the sample (the tomato) was placed. Each light in the control section was adjusted to produce a uniform light distribution for a better capture view. This was done to influence the captured images' quality and prevent other environmental factors, such as weather, from interfering with the system's accuracy.

2. *Camera:* The capturing system was accomplished using a KS2A17 high-resolution camera with a lens zoom range of 18 to 55mm installed on a tripod stand and 38 cm away from the sample. The photograph was taken at Changchun University of Science and Technology's Machine Vision & Robotics Lab. Images of four different example classes were captured at angles of 0, 45, 90, and 135 degrees to cover all possible sample surfaces (top, bottom, and sides1 and 2).

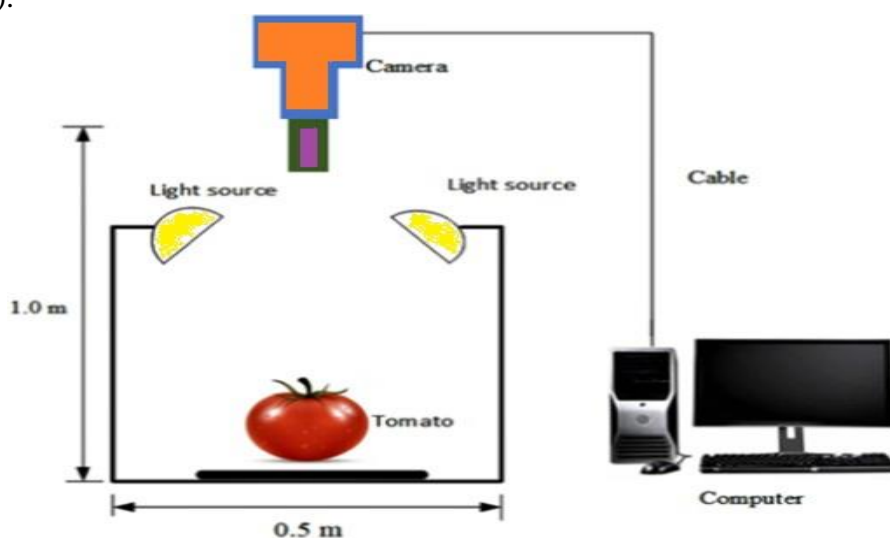


Figure 2: Experimental Setup [11, 14]

3. *Gabbler:* This background served as a desk for the samples to be placed on. It was first covered with a white cloth to capture 200 samples, then with a black cloth to create a dark background, onto which 300 tomato samples were later placed and captured.

4. *PC:* The capture and storage were done on a 2.9GHz Core i5 13" personal computer. It was used to download an image acquisition tool package in MATLAB to capture high-resolution cameras with a pixel value of 1080*1800.

5. This is the subject of an investigation. In this case, cherry tomatoes in two stages of ripening (light red and red) were used, along with four varieties of defects (spot, BER, calyx, and non-calyx). Five hundred (500) samples were collected and labeled as a training and test dataset before being processed and analyzed to solve the classification problems.

3.3 Developed Framework Model

Developing an effective and efficient algorithm for classifying and grading different varieties of tomato fruits concerning their anomalies is the most significant stage in the realization of an automated sorting system. They aid in reducing classification and grading problems, labor costs, and time usage. The research was done to come up with a more effective and efficient way to grade and sort healthy and damaged (spot, BER, calyx, and non-calyx) cherry tomatoes based on their quality. Five hundred samples (500) were captured and used to form a training and testing dataset designed to undergo five different phases: acquisition, pre-processing, feature extraction and selection, classification, and result analysis using appropriate algorithms. Image processing included sharpening, adding noise, and filtering. Next, a process called segmentation was used to get a binary image with the region of interest (ROI). Finally, important unique characteristics called “features” were extracted. The RGB and HSV color spaces, the Textural Gray Level Co-occurrence Matrix (GLCM), the statistical mean, standard deviation, skewness, variance, and defected region area of both images were all calculated and analyzed. The assessment was experimented with using supervised learning algorithms comprising both deep neural networks (DNN) and deep convolutional neural networks (DCNN). Firstly, a deep neural network algorithm comprised of 10 hidden layers was designed and used to perform the grading and classification via MATLAB 2018a software. The neural network uses selected features as an input into the pattern classification, and the output is denoted with two binary digits (00, 01), representing defective and good. Secondly, a deep convolutional neural network was explored using a pre-trained Alex net network. In this regard, feature learning was observed successfully executing convolution, RELU, pooling, and classification. With the use of the Alex net, the image sizes were resized uniformly to 227*227 as an input image. From the series of convolution layers to the classification layer, there are unlimited filters that automatically make the system learn features and then execute classification with higher accuracy. Fig. 3 below gives a descriptive overview of the adopted methodology.

3.4 Algorithms

1. For all the samples N do;
2. Resize the image to 227*227*3
3. Augment @ 0°, 45°, 90° & 135° for a whole surface coverage.
4. A salt & pepper noise was added to the obtained grey images.
5. A 3*3 median filter function was adopted to obtain a definite image
6. The segmentation was performed using an improved Otsu algorithm.

$$G_{(i,j)} = \begin{cases} 1 & (i,j) \geq T \\ 0 & (i,j) < T \end{cases} \quad (1)$$

Improved Otsu Thresholding Algorithm

1. Threshold the initial mean grey value and set it to T_0 .
2. Partition T_0 into two classes: ($W_1 = 0, 1, 2, \dots, T$; $W_2 = T + 1, T + 2, \dots, 255$)
3. Compute low and high threshold values $\{T_1, T_2\}$ respectively.
4. Calculate the inter and intra class variance appropriately for $\{T_1, T_2\}$.
5. Determine the minimum variance ratio.
6. Select the optimal threshold to obtain the binary image.
7. If successful, apply equations (2) & (3) and go to 15; else, repeat steps 8-13.
8. $C \diamond D = \{(C \ominus D) \oplus D\}$ (2)
9. $C \oplus D = \{Z | (\hat{D})Z \cap C\}$ (3)
10. Opening, filling & dilation were all applied with (2 & 3) for a smaller pixel of holes to get ROIs.
11. For a defective sample, the contour of the defective ROI was extracted; else, go to (13).

12. Repeat all (1) to (12) for all samples 1: N.
13. Stop

4. Features Extraction

Features analysis deals with the extraction, selection, and normalization of features. Features extraction usually helps reduce the number of dimensions of perfectly segmented images into a new feature space. This makes classification easier, saves time and space, and improves accuracy [15]. The feature selection subsequently chooses a subset of relevant features from the generated feature set and removes an irrelevant and redundant feature without any alteration for better classification. From the reviewed literature, lots of feature analysis algorithms have been introduced and implemented [11], [16], [17], and [18]. But for this problem, four essential features have been considered with their respective algorithms developed: color, statistical GLCM, texture, and defect.

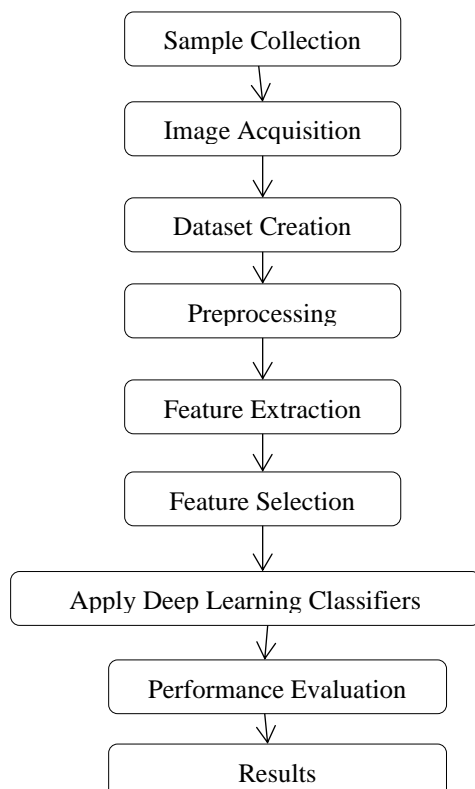


Figure 3: Flowchart [2]

4.1 Statistical GLCM

The Gray Level Co-occurrence Matrix (GLCM) is a second-order statistical feature introduced by Haralick in 1973 for extraction purposes [19]. It works based on the co-occurrence matrices, with a distinctive gray pixel value equal to the same number of rows and columns, respectively. GLCM is derived from the pixel's distance and their relative angular orientation (d, θ). It initially consists of fourteen features, all of which are promising, but only four play a significant role in addressing these assessment issues with a defined spatial relationship ($d = 1$) and ($\theta = 0^\circ, [45]^\circ, [90]^\circ, [135]^\circ$) and a probability density function of (P_{ij}) which calculates pixel pair intensity values (i, j). The extracted feature vectors were derived from homogeneity, energy, contrast, and correlation and performed excellently well in computing our feature vector [20].

The statistical feature formula representation for some of the significant features is listed

below in Table 2. These formulas are with reference to [21].

Table 2:List of Significant Statistical Features Formula [21]

Feature	Formula
Mean (m)	$m = \sum_{i=0}^{L-1} i p(i)$
Variance	$\mu = \sum_{i=0}^{L-1} (i - m)^2 p(i)$
Skewness	$Skewness = \sum_{i=0}^{L-1} (i - m)^3 p(i)$
Kurtosis	$Kurtosis = \sum_{i=0}^{L-1} (i - m)^4 p(i)$
Contrast	$Contrast = \sum_{i=1}^A \sum_{j=1}^A (i - j)^2 p(i, j)$
Correlation	$Correlation = \frac{\sum_{i=1}^A \sum_{j=1}^A (i - m1) (j, m2) p(i, j)}{\sigma_1 \sigma_2}$
Energy	$Energy = \sum_{i=1}^A \sum_{j=1}^A (p(i, j))^2$
Homogeneity	$Homogeneity = \sum_{i=1}^A \sum_{j=1}^A \frac{p(i, j)}{1 + i - j }$
Entropy	$Entropy = \sum_{i=1}^A \sum_{j=1}^A p(i, j) \log(p(i, j))$
Maximum probability	$Maximum\ probability = MAX_{i,j} p(i, j)$

In image processing, feature extraction is an excellent method for dimensionality reduction. When the input data to an algorithm is considered to be processed and most of the received input data does not have exact information, the input data will be converted into a reduced representation of features (a features vector explicitly). Converting the input data into a well-formed set of features is called “feature extraction.”

4.2 Normalization

Normalization of features is usually employed immediately after the feature's extraction or selection before further analysis due to its substantial effect on network performance. The generated vector features are based on many different techniques and algorithms. As such, some variation needs to be individually normalized, column- or row-wise [22]. Firstly, to reduce the unwanted variation between each set and subsequently consider the entire set, the data on different scales is to be compared by converting them to a predefined range of unified scales. Though numerous scaling techniques exist: min-max, z-score, SoftMax scaling [23], [24], etc., But Min-Max techniques have been implemented and have been discussed as follows:

4.2.1 Min-Max technique

It is a predefined technique that offers a linear transformation on the original range of data between 0 and 1 or -1 to 1. For the generation of a similarity relationship between all the extracted and selected feature vectors, this methodology was implemented via the use of the following mathematical expression:

$$\bar{x}_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} (\max(x_{new}) - \min(x_{new})) + \min(x_{new})$$

The \bar{x}_i indicates the normalized derived value, x_i gives a particular row or column features, and $\min x_i$ and $\max x_i$ represent the respective minimum and maximum features among the entire extracted features. At the same time, $\min x_{new}$ and $\max X_{new}$ denote the new minimum and maximum feature values for this technique, which are equal to 0 and 1, respectively.

5. Classification and Prediction Analysis

5.1 Neural Network & Implementation

An artificial neural network is popularly referred to as a "neural network" and is structured in analogy with the human brain, which involves the connections of approximately 10^{11} neurons based on these three essential elements: dendrites, synapses, and axons, respectively [16] and [25]. The dendrites, the input, are responsible for signal transmission into the system through a unit termed the axon (hidden layer), while the synapse presents the output. Similarly, ANN can be typically designed with a fundamental layer, which comprises the input and output layers, and rarely with the inclusion of an optional layer called the hidden layer, which leads to the production of a deep neural network, as illustrated in Figure 4. The input layer took in the extracted image features, passed them to the hidden layer in a forward pattern for necessary processing, and then produced the resultant in the output layer based on the prior label and the number of classes. It has been used for prediction, classification, and recognition, among others.

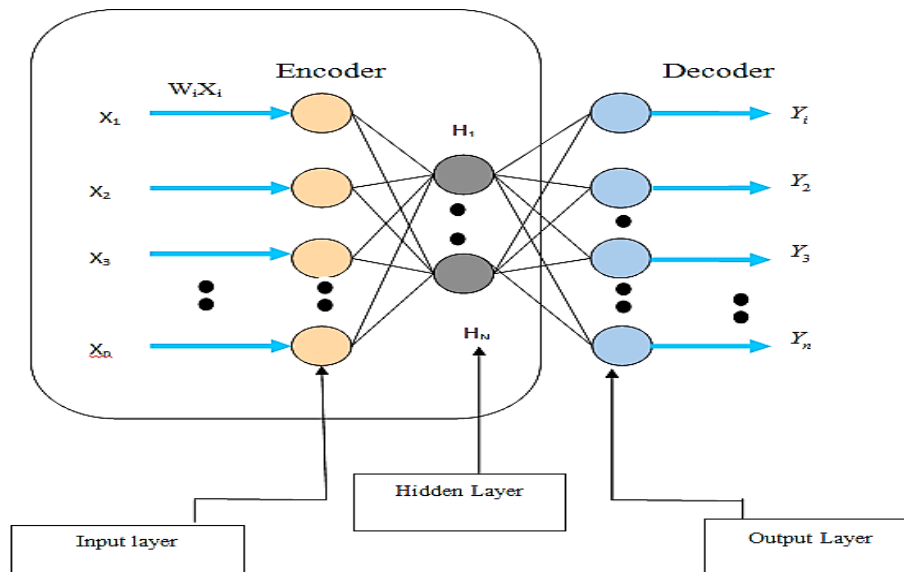


Figure4:Neural Network Architecture [4]

$$y = f\left(\sum w_i x_i + b\right)$$

$$y = f\left(\sum w_i x_i + b\right)$$

where y =target, f = transfer function, w_i = weight, x_i = input and b =bias.

5.1.1 Multi-layer perceptron

In this work, the log-sigmoid function was used in the hidden layer to get a weighted sum of its inputs and a hard limit in the output layer so that 0s and 1s could be separated into two classes. It has been successfully used for self-learning and has been able to perform more

accurately with greater classification efficiency. The various weights in the designated network are updated via the implementation of the following equations:

$$w_{ij}(x+1) = w_{ij}(x) + \nabla w_{ij}(x)$$

$$\nabla w_{ij}(x) = \eta * y_i(x) * \delta_j(x)$$

$$\delta_j(x) = y_j(x) * (1 - y_j(x)) * EJ(x)$$

$$y_j(x) = \frac{1}{1 + E^{-x}}$$

η = learning rate; $\delta_j(x)$ = gradient error ;

$y_j(x)$ = logsigmoid activation; $e_j(x)$ = network error

6. Results and Discussion

The design and modeling analysis were performed on Matlab 2018a with a single Intel Core i5 central processing unit (CPU) with the following specifications: 4 GB of memory, a 480 GB solid-state drive, and 2.9 GHz processor speed. Two distinct supervised learning architectures, one deep neural network and the other convolutional neural network, were effectively modeled. The performance evaluation metric was applied to both models separately to record their success or failure in the process of quality detection, assessment, and classification. A benchmark was then established based on the results of the comparative analysis. As a result, the results of each model are presented in tables and graphs after some critical steps.

Table 3: DNN Hyperparameter Declaration

Node	Layers	A	E	L.R
1	10	Sigmoid & Linear	24	0.0001

Keys: A=activation, E=epoch, L.R= learning rate

6.1 DNN Model of Experimental Analysis

The DNN model, derived from an ANN, was created by selecting and extracting a significant and irredundant set of features. These features include the mean, variance, standard deviation, and skewness of the RGB and HIS color spaces and their conversion, the contrast, correlation, energy, and homogeneity of the GLCM, and the estimation of the defective proportion with an area and perimeter. This equates to the number of 30 features used as input to the input layer. The preceding layer was connected to a single-node hidden layer constructed with ten (10) neurons and successfully coupled to an output layer along with a linear transfer function to suit the linear classification purpose for the two defined classes (good or defective), yielding two at the output.

6.2 DNN Training Analysis

The dataset was fundamentally segmented into training and testing categories using the 500 captured samples. To improve system performance, reduce error rates, and achieve the ideal fit, the obtained samples were supplemented with rotation and translation techniques, yielding a total of 800 samples. These samples (tomatoes) were used to train the network, with a training percentage of 70% (560 samples) and testing and validation percentages of 15% (120 samples) each. The cross-entropy (CE) obtained for each training percentage was kept to a bare minimum and indicated an appropriate classification due to the inverse relationship: the lower the values, the better the output. The percentage error (percent E) stated the proportion of misclassified samples, which was also negligible in this regard. A value of zero for CE and percent E indicates the absence of error and misclassification. In contrast, a value of 100 indicates the maximum.






	 Samples	 CE	%E icon" data-bbox="690 88 710 102"/> %E
 Training:	560	1.52055e-0	2.50000e-0
 Validation:	120	4.23583e-0	8.33333e-1
 Testing:	120	4.32561e-0	4.58333e-0

Figure 5: Training, Validation, and Testing Breakdown [18]

The model was initially designated to undergo 14 iterations within a zero-epoch duration with performance, gradient, and validation checks of 0.00967, 0.0258, and 6. Hence, the performance state was re-evaluated mostly with the mean squared error (MSE) and the number of epochs in the course of training. The MSE gives the average squared difference between outputs and estimated targets. The lower the values become, the better the system's performance, as zero MSE indicates no error. MSE for the three processes (training, validation, and testing) has been illustrated in Figure 5 with blue, green, and red, respectively. At the initial stage of the training exercise, the observed MSE for the three classes originated from 10^0 , which is equivalent to 1 at the exact zero epoch of iteration. As the training progressed, there was a significant drop in the MSE with an increase in the number of epochs until an ideal fit was reached with an ultimate validation performance of 0.0236 drawn with a dotted light green line at an epoch of 8 out of an overall 14 epochs.

6.3 DNN Evaluation Metric Analysis

The evaluation metric was carried out using the obtained confusion or error matrix. Based on row and column designations, it summarized all predicted class (target) instances versus the actual class (output) in a matrix format. While the true positive (TP) and true negative (TN) were both represented, the false negative (FN) and false positive (FP) were not. TP and TN produced accurate predictions, whereas FN and FP produced misclassifications.

However, for the DNN analysis, a confusion metric was obtained for the three experimental classes: training, validation, and testing. The overall confusion matrix, termed "all confusion matrix," was derived from them, which gives the success or failure of the overall classification accuracy. As a result, the confusion metric has been successfully used to evaluate the designated DNN model's performance with the computation of recall or sensitivity of true positive rate (TPR), specificity of true negative rate (TNR), F1 score, precision, and accuracy, as shown in the breakdown tables below. Furthermore, the confusion matrix summarized the three previous cases: with an overall sample count of 800, 790 were correctly classified as true positives, and 10 samples were misclassified as false negatives. No sample was classified as both a false positive and a true negative. The evaluation metric derived from the all-confusion matrix section of the tables is presented in the figures below, with a probability and percentage accuracy of 0.988 and 98.8 percent for the true positive and negative cases, respectively.

Table 4: DNN All Confusion Analysis

Analysis	Formula	Substitution	Result
Recall (R)	$TP / (TP + FN)$	$790 / (790 + 10)$	0.988
Precision (P)	$TP / (TP + FP)$	$790 / (790 + 0)$	1
Specificity	$TN / (TN + FP)$	$0 / (0 + 0)$	0
F1score	$(2 * P * R) / (P + R)$	$2 * 1 * 0.988 / (1 + 0.988)$	0.994
Accuracy	$(TP + TN) / (TP + TN + FP + FN)$	$790 / (790 + 10)$	0.988

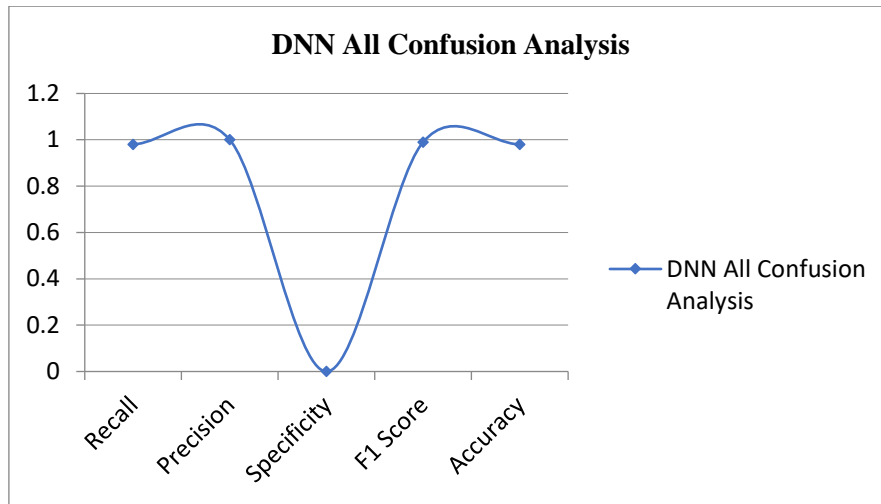


Figure6:DNN All Confusion Analysis

6.4 CNN Model of Experimental Analysis

The CNN architecture used a fine-tuned, pre-trained Alex Net with 57 million parameters and a tomato image scaled to 227*227*3. The first five layers are convolution layers, while the last three are fully connected layers. The former was used for feature extraction and learning. At the same time, the latter was utilized for classification, with the output falling into classes 1 or 2. As shown in Table 10, we used the stochastic gradient descent (sgdm) activation function with a learning rate of 0.0001, a maximum of 20 epochs, and a minimum of 64 batches. The training and testing datasets were correctly randomized at 70:30 ratios, respectively.

Table 5: CNN Hyperparameters Declaration

Mini-batch	Activation	Epoch	Learning Rate
64	ReLU&Sgdm	20	0.0001

6.5 CNN Confusion Matrix

The overall confusion matrix was also acquired for this scenario, and it has been used to derive the other metric analyses as presented in the following tables below.

Table 6: DNN All Confusion Analysis

Analysis	Formula	Substitution	Result
Recall (R)	$TP / (TP + FN)$	$716 / (716 + 15)$	0.979
Precision(P)	$TP / (TP + FP)$	$716 / (716 + 4)$	0.994
Specificity	$TN / (TN + FP)$	$65 / (65 + 4)$	0.942
F1score	$(2 * P * R) / (P + R)$	$2 * 0.994 * 0.979 / (0.994 + 0.979)$	0.986
Accuracy	$(TP + TN) / (TP + TN + FP + FN)$	$(716 + 65) / (716 + 65 + 4 + 15)$	0.976

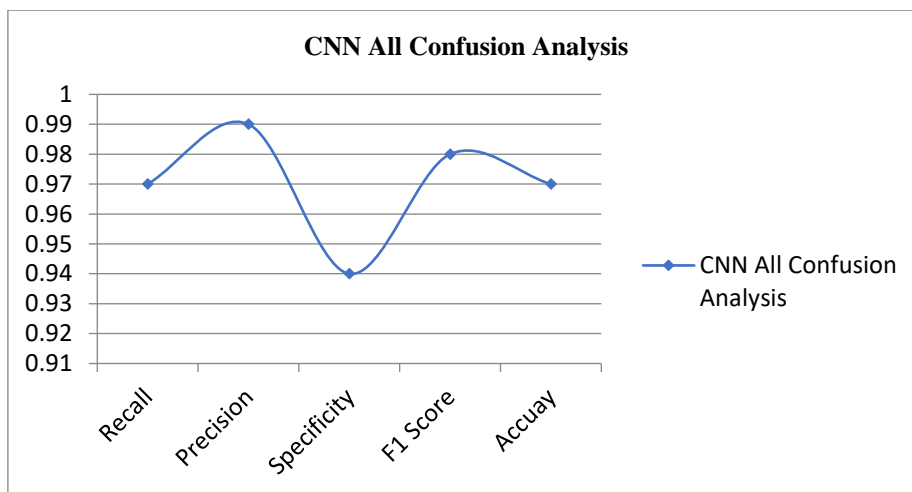


Figure7:CNN All Confusion Analysis

6.6 DNN& CNN Evaluation and Comparative Analysis

It is important to note that the two implemented supervised learning architectures for deep and convolutional neural networks have been adequately assessed and benchmarked to give future recommendations. The table shows the estimated overall accuracy, activation function, and maximum number of epochs reached.

Table 7:DNN and CNN Evaluation and Comparative Analysis

Network Architectures		
Parameters	DNN	CNN
Activation	Log sigmoid & Linear	ReLU&Sgdm
Epoch	14	20
Accuracy	98.8%	97.6%
Precision	1	0.994
Specificity	0	0.942
Recall	0.988	0.979
Loss	1.2%	2.4%
F1score	0.994	0.986
Duration	8m 12s	24m 16s

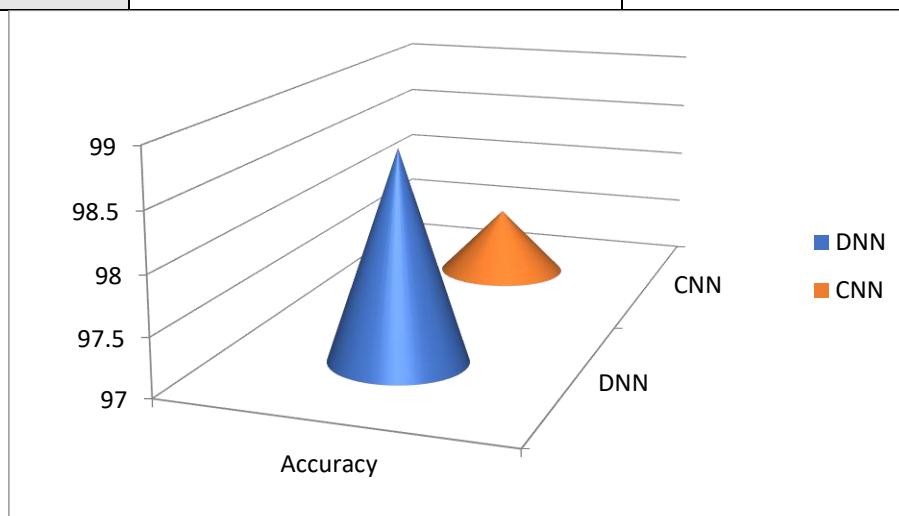


Figure8:Comparative analysis of DNN and CNN

The analytical results for both architectures were great. Still, DNN outperformed CNN by 1.2 percent and reduced the loss function by half. The maximum iterations were 14 with unity precision and zero specificity, and the CNN was 0.994 and 0.942.

6.7 Comparison of Accuracy of Proposed Model with Previous Work

In Section 2, we provided some previous works and their accuracy levels. The accuracy level is given in Table 8.

Table 8: Comparison of Accuracy with Previous Work

Previous Study	Dataset Used	Accuracy Level
Mohit Agarwal et al. [6]	PlantVillage	Traditional ML approach 98.4 % accuracy. With k-NN 93.5 % accuracy
M. S. Irajji [9]	Tomato Pictures from Farm	83.2 % Sensitivity 96.5 specificity 89.40% g-mean 95.5 % accuracy
P. Wan et al. [10]	Tomato Pictures	Standard Deviation of 1.2 % and accuracy 99.31 %
L. C. Santosh et al. [11]	Tomato Pictures	These models have a discriminating accuracy of 100% /85% and 80% /96% with some sensitivity cultivars, with cross-validation and prediction errors of 7%, respectively.
N. El-Bendary et al. [13]	230 Samples of Tomato Pictures	92.72 % accuracy using a linear kernel function.
Proposed Model	Cherry Tomatoes from Local market, Hyderabad	DNN evaluation 98.8 % and CNN evaluation 97.6 %

7. Conclusion and Future Scope

The work has been carried out to evaluate its external and internal characteristics through two different supervised learning methods. The training and analysis data sets are based on four different types of tomatoes (spot, BER, calyx, and non-calyx). A deep neural network with ten hidden layers was built. A refined neural network of eight key layers from Alex Net Architecture was utilized separately to measure and classify MATLAB 2018a software. Thus, both networks carried out quality control. They performed exceptionally well in solving both classes, with approximately 97% more precise predictability than the prediction accuracy of a small loss function in both situations. However, the algorithms can be used in the agriculture industry to detect, sort, and discover faults in fruits or vegetables of all kinds. However, since CNN was the only one who performed a real-time check automatically, DNN was unable to complete it.

Furthermore, the number of samples collected was limited. These designs did not operate effectively until they were later extended based on a rotating approach and translation to enhance their performance and ability to predict. It may be concluded that the two networks have been correctly implemented and evaluated in good and defective classes for classifying South Indian cherry tomatoes.

Future Work: Based on the results of the experimental investigation with DNN and CNN architectures, future researchers must fill in several gaps. These include using a more extensive training dataset to avoid underfitting issues and the requirement for additional data augmentation. To widen the scope of the investigation, other fruit species and illness or defect variants should be explored. Our established network algorithms can be integrated with a robotics machine for more visible realization and higher-quality evaluation.

References

- [1] H. Sabrol and S. Kumar, "Fuzzy and Neural Network-based Tomato Plant Disease Classification using Natural Outdoor Images," *Indian Journal of Science and Technology*, vol. 9, no. 44, pp. 1107-1114, May 2016, DOI: 10.17485/ijst/2016/v9i44/92825
- [2] M. H. Malik, T. Zhang, H. Li, M. Zhang, S. Shabbir, and A. Saeed, "Mature Tomato Fruit Detection Algorithm Based on improved HSV and Watershed Algorithm," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 431-436, Sep. 2018. <https://doi.org/10.1016/j.ifacol.2018.08.183>
- [3] N. Vélez Rivera et al., "Early Detection of Mechanical Damage in Mango using NIR Hyperspectral Images and Machine Learning," *Biosystems Engineering*, vol. 122, no. 1, pp. 91-98, May 2014. DOI: 10.1016/j.biosystemseng.2014.03.009
- [4] H. J. G. Opena and J. P. Yusiong, "Automated Tomato Maturity Grading Using Abc-Trained Artificial Neural Networks," *Malaysian Journal of Computer Science*, vol. 30, no. 1, May 2017, DOI: <https://doi.org/10.22452/mjcs.vol30no1.2>
- [5] M. Brahim, K. Boukhalfa, and A. Moussaoui, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299-315, 2017, DOI: 10.1080/08839514.2017.1315516
- [6] Mohit Agarwal, Suneet Gupta and K. K. Biswas, "Development of Efficient CNN model for Tomato crop disease identification," *Sustainable Computing: Informatics and Systems*, vol. 28, no. 1, p. 100407, Dec 2020, DOI: 10.1016/j.suscom.2020.100407
- [7] Reem Mohammed Jassim Al Akkam, Mohammed Sahib Mahdi Altaei, "Plant Leaf Diseases Detection using Deep Learning," *Iraqi Journal of Science*, vol. 63, no. 2, pp. 801 – 816, April 2021. DOI: 10.24996/ijis.2022.63.2.34
- [8] N. Alavi, "Quality Determination of Mozafati Dates using Mamdani Fuzzy Inference System," *Journal of the Saudi Society of Agricultural Sciences*, vol. 12, no. 1, pp. 137–142, 2013, <https://doi.org/10.1016/j.jssas.2012.10.001>
- [9] S. Munera, J. Blasco, J. M. Amigo, S. Cubero, P. Talens, and N. Alexios, "Use of Hyperspectral Transmittance Imaging to Evaluate the Internal Quality of Nectarines," *Biosystems Engineering*, vol. 182, no.1, pp. 54-64, April 2017, DOI: <https://doi.org/10.1016/j.biosystemseng.2019.04.001>
- [10] Iraj, M. S. "Comparison between Soft Computing Methods for Tomato Quality Grading using Machine Vision," *Journal of Food Measurement and Characterization*, vol. 2018, no. 13, pp. 1-15, Aug. 2018, DOI: <https://doi.org/10.1007/s11694-018-9913-2>
- [11] P. Wan, A. Toudeshki, H. Tan, and R. Ehsani, "A Methodology for Fresh Tomato Maturity Detection using Computer Vision," *Computers and Electronics in Agriculture*, vol. 146, pp. 43-50, Feb. 2018, <https://doi.org/10.1016/j.compag.2018.01.011>
- [12] L. C. D. Santosh Shrestha, Merete Halkjær Olesen and René Gislum, "Use of Multispectral Imaging in Varietal Identification of Tomato," *Sensors*, vol. 15, pp. 4496-4512, Feb. 2015, <https://doi.org/10.3390/s150204496>
- [13] G. Liu, S. Mao, and J. H. Kim, "A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis," *Sensors*, vol. 19, no. 9, pp. 20-23, Apr. 2019, Doi: 10.3390/s19092023
- [14] N. El-Bendary, E. El Hariri, A. E. Hassani, and A. Badr, "Using Machine Learning Techniques for Evaluating Tomato Ripeness," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1892-1905, Oct. 2015, <https://doi.org/10.1016/j.eswa.2014.09.057>
- [15] D. Ileri, E. Belal, C. Okinda, N. Makange, and C. Ji, "A Computer Vision System for Defect Discrimination and Grading in Tomatoes using Machine Learning and Image Processing," *Artificial Intelligence in Agriculture*, vol. 2, pp. 28-37, Jun. 2019, <https://doi.org/10.1016/j.aiaa.2019.06.001>
- [16] H. G. Amir Alipasandi and Saman Zohrabi Alibeyglu, "Classification of three Varieties of Peach Fruit Using Artificial Neural Network Assisted with Image Processing Techniques," *International Journal of Agronomy and Plant Production*, vol. 4, no. 9, pp. 2179-2186, Jan. 2013.

- [17] A. Ren et al., "Machine Learning-Driven Approach Towards the Quality Assessment of Fresh Fruits Using Non-Invasive Sensing," *IEEE Sensors Journal*, vol. 20, no. 4, pp. 2075-2083, Oct. 2020, <https://doi.org/10.1109/JSEN.2019.2949528>
- [18] S. Sabzi, Y. Abbaspour-Gilandeh, and G. García-Mateos, "A New Approach for Visual Identification of Orange Varieties using Neural Networks and Metaheuristic Algorithms," *Information Processing in Agriculture*, vol. 5, no. 1, pp. 162-172, Mar. 2018, <https://doi.org/10.1016/j.inpa.2017.09.002>
- [19] O. R. Indriani, E. J. Kusuma, C. A. Sari, E. H. Rachmawanto, and D. R. I. M. Setiadi, "Tomatoes Classification using K-NN based on GLCM and HSV Color Space," in *proceedings of International Conference on Innovative and Creative Information Technology (ICITech)*, pp. 1-6, Salatiga, Indonesia, Nov. 2017.
- [20] V. Pavithra, R. Pounroja, and B. S. Bama, "Machine Vision-Based Automatic Sorting of Cherry Tomatoes," in *proceedings of 2nd International Conference on Electronics and Communication Systems (ICECS)*, pp. 271-275, Feb. 2015, DOI:10.1109/ECS.2015.7124907, Coimbatore, India
- [21] Massimiliano Bonamente, *Statistics and Analysis of Scientific Data*, Newyork: Springer Science+ Bussiness Media, 2017.
- [22] H. Zhao and X. Zhou, "Recognition of Artificial Ripening Tomato and Nature Mature Tomato Based on the Double Parallel Genetic Neural Network," *Advance Journal of Food Science and Technology*, vol. 5 no. 4, pp. 482-487, Apr. 2013, <http://dx.doi.org/10.19026/ajfst.5.3295>
- [23] B. K. Singh, K. Varma and A. S. Thoke, "Investigations on Impact of Feature Normalization Techniques on Classifier's Performance in Breast Tumor Classification," *International Journal of Computer Applications*, vol. 116, no. 19, pp. 11-15, Apr. 2015.
- [24] L. Xie, Q. Tian and B. Zhang, "Simple Techniques Make Sense: Feature Pooling and Normalization for Image Classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 7, pp. 1251-1264, July 2016.
- [25] Y.-C. Du and A. Stephanus, "Levenberg-Marquardt Neural Network Algorithm for Degree of Arteriovenous Fistula Stenosis Classification Using a Dual Optical Photoplethysmography Sensor," *Sensors*, vol. 18, no. 7, pp. 2322-2331, July 2018, Doi: 10.3390/s18072322