RESEARCH ARTICLE

Deep Learning-based Classifications of Fresh and Rotten Fruits using Convolutional Neural Network

Muhammad Idrees[1], **Altaf Hussain**[1,*], **Nasir Hussain**[1], **Usman J. Wushishi**[1], **Muhammad Imran**[1] **and Atif Ali Wagan**[1]

[1] School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract

Global food waste has increased due to inefficiencies in production, processing, and distribution, making food security an increasingly critical issue. Rising global food demand and substantial losses along the food supply chain emphasize the need to optimize available resources for effective food management. Spoilage detection matters because a rotten fruit left among sound ones can spread decay and cause further losses. Sorting is still largely done by hand, which is slow, costly, and inconsistent between inspectors. This study proposes a deep learning method that identifies the fruit type and classifies it as fresh or rotten from a single image, aiming to reduce manual sorting effort and cost. Convolutional Neural Networks (CNNs), with their ability to automatically extract hierarchical features from images, are well-suited for distinguishing between fresh and rotten fruits. In our approach, convolutional layers detect patterns and edges, while pooling layers reduce feature dimensionality to improve classification efficiency. The enhanced CNN, trained with data augmentation on 13,568 images of fresh and rotten apples, bananas, and oranges (six classes), identifies the fruit type and predicts its freshness in one pass. Experimental results show a validation accuracy of 99.03% and a training accuracy of 99.46%, demonstrating the effectiveness of the proposed method. The results highlight the potential of deep learning-based automated systems in agricultural and retail settings, where they can improve fruit quality assessment, ensure consumer safety, and reduce operational costs. This work underscores the importance of integrating machine learning technologies into food production and supply chains to enhance efficiency and sustainability.

Keywords: fresh and rotten fruits, feature extraction, fruit, neural network, deep learning, convolution neural network (CNN), fruit classification.

1 Introduction

Recent advances in computer vision, particularly through machine learning (ML) and deep learning (DL), have substantially improved the performance of image classification tasks [1–5]. One of the

Submitted: 19 August 2025
Revised: 07 November 2025
Accepted: 25 December 2025
Published: 15 November 2026

Vol. 2, **No.** 4, 2026.
10.62762/TACS.2025.384927

***Corresponding author:**
✉ Altaf Hussain
altafkfm74@gmail.com

Citation
Idrees, M., Hussain, A., Hussain, N., Wushishi, U. J., Imran, M., & Wagan, A. A. (2026). Deep Learning-based Classifications of Fresh and Rotten Fruits using Convolutional Neural Network. *ICCK Transactions on Advanced Computing and Systems*, 2(4), 272–290.

© 2026 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

critical challenges in the agricultural sector is the detection, classification, and sorting of imperfect produce, including its separation into fresh and spoiled categories [6–10]. Fresh produce is exposed to microbial contamination [11], and spoiled fruits that are not separated from fresh ones can further reduce overall productivity and quality; automated fresh/rotten classification has therefore been studied [12]. Traditionally, this sorting process is performed manually, which is labor-intensive, time-consuming, and prone to human error, leading to increased production costs and inefficiencies [1, 2]. Consequently, there is a growing demand for automated systems that can enhance efficiency, reduce labor requirements, and minimize operational costs while maintaining high accuracy.

Image classification plays a pivotal role in numerous domains, including video analysis, facial recognition, and object detection [13–15]. Deep learning, a subfield of machine learning, has demonstrated superior performance in image recognition due to its ability to automatically extract hierarchical features through multi-layered network architectures [13, 15], and CNN-based models have achieved high accuracy in agricultural image recognition tasks such as crop disease identification [16–20]. This capability yields improved accuracy and robustness compared to traditional ML approaches, which rely heavily on manually designed features [4]; this advantage is reflected in the growing use of CNNs for fruit image processing [3]. In recent years, logistics, supply chain management, and agricultural sectors have begun leveraging DL for practical applications, including crop nutrient and disease monitoring [21–23], farm and supply-chain data analytics [24], and fruit detection and yield estimation [25, 26]. By extracting visual features and applying CNN-based classification [13, 27], DL models can classify fruit images reliably [28] and distinguish between fresh and spoiled produce [8], while image-based defect detection [5] supports quality control and reduces dependence on manual inspection.

Machine learning and deep learning have been widely adopted in business and industrial processes, such as autonomous vehicles, virtual assistants, and video surveillance, highlighting their transformative potential [14, 29]. Within agriculture, these technologies contribute to "smart farming" initiatives, enabling data-driven decision-making through in-field data collection, processing, analysis, and forecasting [7, 24]. Such systems improve crop yield, resource utilization, and adaptability to changing environmental conditions [24]. Among these applications, automated fruit classification has emerged as a particularly impactful use case, offering both operational efficiency and enhanced product quality [30–36].



Figure 1. Components of the smart farming system for fruit classification, including image acquisition, preprocessing, CNN-based feature extraction and classification, and decision-making for quality sorting.

Figure 1 illustrates several key components of smart farming. For instance, fruit classification systems have harnessed the powerful automation and computational capabilities of deep learning and machine learning to analyze and predict farm yield, as well as to detect diseases and defects in the field.

Consumers typically make rapid purchasing decisions about fruits based solely on visual appearance. It is virtually impossible for an average person to determine with absolute certainty whether a given fruit is good or sweet simply by examining its species and genetic traits. Nevertheless, the outcomes of such basic visual assessments can reflect the perceived quality of the fruit, which implies that suppliers and retailers must ensure that the fruits offered for sale appear attractive to boost sales. This is why grocery stores and supermarkets invest considerable effort in inspecting and sorting their fruit inventory. Certified inspectors evaluate and grade fruits for quality assurance and health reasons. However, inconsistencies in their reports can lead to substantial waste or poor purchasing decisions. If spoiled fruits

are misclassified as fresh, or vice versa, consumers may make suboptimal choices. Large quantities of food are lost or wasted along the food supply chain [37]. An additional benefit of fruit identification is that it shortens the supply chain by reducing the time and energy spent on sorting fruits at retail outlets. Automating the sorting process and eliminating improper handling by non-professionals could also help mitigate food safety issues, particularly in an era where new viral and bacterial strains pose ongoing health challenges [11].

In this article, we aim to develop a highly accurate system for fruit classification that distinguishes between various fruit types and determines whether the fruits are fresh or rotten from input images. The proposed solution involves training an optimized CNN model, which achieves a remarkable training accuracy of 99.46% and a validation accuracy of 99.03%, demonstrating strong generalization capability. The dataset used for training consists of approximately 10,880 fruit images, with an additional 2,688 images for testing, all categorized into six distinct classes.

The key contributions and novel aspects of this work are as follows:

- The CNN model is constructed with four convolutional layers, progressively increasing the number of filters from 32 to 256, to efficiently extract hierarchical features from the input images.
- To address the challenge of limited data, a robust data augmentation strategy has been implemented. In addition to traditional augmentations such as rotation, flipping, zooming, and brightness adjustments, we introduce novel techniques including random cropping (to introduce spatial variability) and channel shifting (to enhance color robustness). These enhancements ensure that the model is trained on diverse variations of the data.
- A novel preprocessing function is employed that applies random cropping to extract a 56×56 region from the image, followed by resizing the cropped patch to the target dimensions of 64×64. This ensures consistent input dimensions while improving the model's ability to generalize across spatial variations.
- A learning rate scheduler, specifically the `ReduceLROnPlateau` method, is employed to adaptively reduce the learning rate when the validation loss plateaus. This feature ensures optimal convergence and prevents the model from becoming trapped in local minima.
- The proposed model demonstrates exceptional performance across all six fruit classes, achieving a training accuracy of 99.46% and a validation accuracy of 99.03%. Such high accuracy underscores the model's capability to distinguish between fresh and rotten fruits with minimal error.
- The proposed model is lightweight, efficient, and suitable for real-world applications, such as automated inspection and grading of fruits in grocery stores. Its high accuracy and robust generalization make it an excellent candidate for deployment in commercial environments, saving time and reducing manual effort. By combining these innovations, the proposed system achieves state-of-the-art performance with a high accuracy rate and robust generalization. This research lays the groundwork for future advancements in automated fruit classification systems.

The advent of smart farming has provided fertile ground for expansion in the agricultural industry, and this is essential rather than merely beneficial. Global demand for food is expected to keep rising for decades as population and consumption grow [38], which requires higher yields and less waste along the food supply chain. Therefore, it is crucial to learn how to farm more efficiently and apply that knowledge to boost crop yields. To reach this goal, improved efforts are needed in three primary areas, which smart farming can support because its big-data applications extend beyond primary production to the entire food supply chain [24]:

- Boosting individual farm output of food crops and livestock.
- Guaranteeing the same or even better quality of agricultural products.
- Reducing food waste from farms.

All of these areas represent potential domains where AI could lead to improved outcomes. The goal of this paper is to help businesses reduce costs by minimizing food waste, which also allows retailers to stock their shelves with more desirable products. For this reason, fruit classification is crucial in many commercial contexts. For example, a cashier at a grocery store could use a fruit classification system to better identify

fruit varieties and their corresponding prices. It could also be used to determine which fruit varieties are suitable for consumers with special dietary needs.

The key novelty of the proposed method lies in its enhanced CNN architecture, which is specifically designed to perform simultaneous fruit type identification and freshness detection, enabling a more comprehensive and automated assessment of produce quality. Unlike conventional approaches that often focus solely on either classification or spoilage detection, this model integrates both tasks within a single framework, improving efficiency and reducing the need for separate inspection systems. Additionally, the method incorporates advanced data augmentation techniques to enhance the model's ability to generalize across diverse fruit images, accounting for variations in lighting, angle, size, and texture that commonly occur in real-world agricultural and retail environments. This not only improves prediction accuracy but also ensures consistent performance under varying conditions. Another important contribution is the focus on reducing human labor and operational costs while maintaining high reliability, addressing practical challenges faced by farmers and retailers in routine fruit quality inspection. Furthermore, the proposed system offers a practical solution for real-time quality assessment, supporting immediate detection of spoiled fruits and preventing contamination of fresh produce—an aspect often overlooked in existing works. By combining these innovations, the framework provides a robust, efficient, and scalable approach for automated fruit quality monitoring, demonstrating both scientific and practical significance in modern agriculture and food supply chains.

The remainder of this article is organized as follows: Section 2 provides a detailed overview of the literature review. Section 3 presents the main method and implementation. Section 4 discusses the results, and Section 5 concludes the article.

2 Literature Review

Machine learning has been applied across numerous sectors, streamlining operations, reducing costs, and optimizing human resource allocation. In agriculture, ML and deep learning (DL) are increasingly being deployed to address pressing global challenges, including food security and crop quality management. Many field operations in crop production are labor-intensive because of their complexity, their contact with delicate edible produce, or their repetitiveness, which further accentuates the need for technological interventions such as agricultural robotics [39].

Deep learning has brought about a paradigm shift in agricultural practices, enabling accurate detection of plant diseases, pest infestations, and defective produce. Wang et al. [5] utilized a machine vision system combined with support vector machines (SVM) to detect surface blemishes on fruits, reporting that SVMs can achieve high accuracy on small datasets, albeit with strong dependence on feature engineering. Palakodati et al. [12] employed a CNN with a Softmax classifier, alongside several transfer-learning baselines, to sort apples, bananas and oranges into fresh and rotten categories on a Kaggle dataset, reporting an accuracy of 97.82% and showing that the custom CNN outperformed the transfer-learning models. Similarly, Ha et al. [16] proposed a precise CNN-based approach for diagnosing Fusarium disease in radishes, while Ma et al. [17] developed a deep CNN to classify four cucumber diseases with an accuracy of 93.41%. Lu et al. [18] applied CNNs to rice disease detection, achieving 95.48% accuracy, further underscoring the superiority of DL over conventional methods.

In a broader study, Mohanty et al. [20] applied CNNs to detect 26 plant diseases across 14 crop species, demonstrating robust generalization and high classification performance. Tran et al. [21] developed a monitoring system for tomato growth and nutrient deficiency identification. Fuentes et al. [22] employed advanced DL architectures, including Faster R-CNN, R-FCN, and single-shot detectors (SSD), combined with powerful feature extractors such as VGG and ResNet, for disease and pest detection in dynamic agricultural environments. Beyond disease detection, a substantial body of work has targeted fruit recognition and freshness assessment directly. Karakaya et al. [40] compared handcrafted descriptors with CNN-based features for classifying three fruit types as fresh or rotten, reporting that CNN-based features consistently achieved the highest accuracy. Mureşan and Oltean [41] introduced the widely used Fruits-360 dataset together with a deep network for large-scale fruit recognition, while Tapia-Mendez et al. [42] extended deep learning to the joint classification and ripeness assessment of fruits and vegetables. Zhang et al. [43] proposed a 13-layer CNN combined with data augmentation for fruit-category classification, and Hou et al. [44] and Lu et al. [45] demonstrated compact CNN architectures for efficient fruit recognition. Complementing these, Bhargava and

Bansal [46] surveyed computer-vision–based quality evaluation of fruits and vegetables, and Marimuthu and Roomi [47] employed a particle-swarm-optimized fuzzy model for banana ripeness classification. Among these works, learned CNN features were reported to outperform handcrafted descriptors for freshness classification [40], and data augmentation was used to improve the generalization of fruit-classification CNNs [43], which motivates the enhanced CNN with data augmentation proposed in this work.

Despite the considerable advances documented in the literature, there remains scope for further improvements in accuracy and computational efficiency through enhanced CNN architectures, coupled with sophisticated data preprocessing, augmentation strategies, and novel feature extraction techniques. The present study addresses this gap by focusing on fruit quality classification, and the proposed methodology is elaborated in the subsequent section.

3 Proposed Methodology

Convolutional neural networks (CNNs) are a specialized class of neural network architectures that are particularly effective for image recognition tasks, owing to their inherent capability to autonomously learn and extract discriminative features from visual data. A typical CNN architecture comprises several key layers, including convolutional layers, pooling layers, and fully connected layers, which collectively enable the network to recognize complex patterns and hierarchical features—such as edges, corners, shapes, and textures—from raw pixel inputs. These features are learned through the backpropagation algorithm, wherein the network iteratively adjusts its weights and biases in response to the prediction error computed for each training sample. As a result, CNNs have demonstrated remarkable effectiveness across a broad spectrum of image recognition applications, ranging from object detection and classification to facial recognition and beyond [13, 15].

Since the first mathematical neuron models were proposed in the 1940s, the field of neural networks has undergone substantial evolution, with numerous novel models and training techniques being introduced. A major breakthrough occurred in the 1980s with the advent of the backpropagation algorithm, which enabled efficient training of neural networks on large-scale datasets and catalyzed significant advances in speech recognition and computer vision. In recent years, the emergence of deep learning methodologies has further expanded the representational capacity of neural networks, allowing them to capture increasingly complex and abstract features from high-dimensional data. Among these, convolutional neural networks—specialized for image and video processing—have achieved state-of-the-art performance on a variety of visual recognition benchmarks, including image classification, object detection, and semantic segmentation [15, 29]. Figure 2 depicts the structure of an artificial neuron unit (node), which serves as the fundamental computational building block of CNNs.

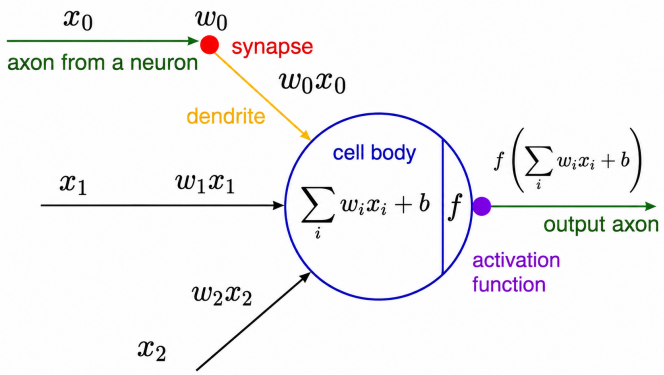


Figure 2. Artificial neuron unit (node), illustrating input signals, weights, bias, activation function, and output, forming the basic computational unit of a CNN.

3.1 Feature Extraction in Neural Networks

David H. Hubel and Torsten N. Wiesel, neurophysiologists who studied the human visual cortex, demonstrated that the visual system perceives the world by first extracting local features from small sections of the incoming image and then combining these extracted features through pooling mechanisms [13]. Our visual system excels at recognizing patterns and extracting features from images, a process carried out by groups of neurons sensitive to specific types of visual input, such as edges, corners, and textures. These neurons collaboratively build a representation of the image, which is subsequently used to interpret its content. Convolutional neural networks, inspired by the structure of the visual cortex, replicate this process by employing layers of neurons trained to recognize distinct features in an image [13].

The curse of dimensionality refers to the difficulty of analyzing and organizing data with a large number of features or dimensions. As the number of features increases, the amount of data required to cover the feature space grows exponentially, making it

increasingly difficult to identify meaningful patterns or relationships. Feature selection techniques aim to reduce the dimensionality of the problem by selecting only the most relevant features, thereby mitigating the curse of dimensionality and improving the efficiency and accuracy of the neural network [27].

3.2 CNN Architecture Components

Multiple hidden layers reside between a convolutional neural network's input layer and its output layer, all intricately interconnected. While a more complex network with additional convolution layers can enhance accuracy, it also increases training time and computational resources while raising the risk of overfitting [48, 49]. Overfitting occurs when the network is tailored so precisely to a specific subset of data that it cannot generalize to new datasets or make reliable predictions. Over time, convolutional neural networks have seen design enhancements, primarily in layer architectures and optimization techniques to improve speed and accuracy. Progress has also been driven by increases in processing power and the volume of annotated data [13]. A typical CNN comprises pooling layers, convolutional layers, a fully connected layer, and an output layer, along with activation functions [13, 15]. The basic architecture of a typical CNN is depicted in Figure 3.

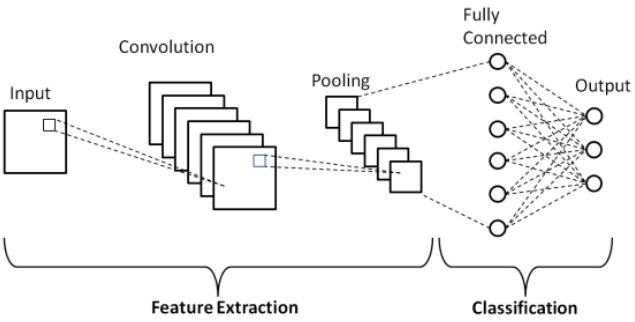


Figure 3. Typical structure of a CNN, showing layers for convolution, activation, pooling, and fully connected classification used for fruit image recognition.

3.2.1 *Convolutional Layer*

Convolutional layers are fundamental to CNN's feature extraction capabilities. Convolution kernels are matrices of image pixels used in image classification applications. By convolving the input with these kernels and applying an activation function, a feature map is generated. The outputs of one layer serve as inputs to the next layer in a chain of multiple such convolutional layers [13]. While adding more convolutional layers can boost accuracy, it also slows down calculations and increases the risk of overfitting [48].

3.2.2 *Pooling Layers*

Pooling layers typically follow convolutional layers. They reduce the network complexity introduced by convolution and mitigate overfitting by decreasing the feature map's height and width. Since the pooling operation is applied to every slice of the input feature map, the depth remains unaffected. Thus, pooling can be viewed as a method of feature selection. The two most common types of pooling are average pooling and max pooling, with max pooling being the preferred choice (see Figure 4).

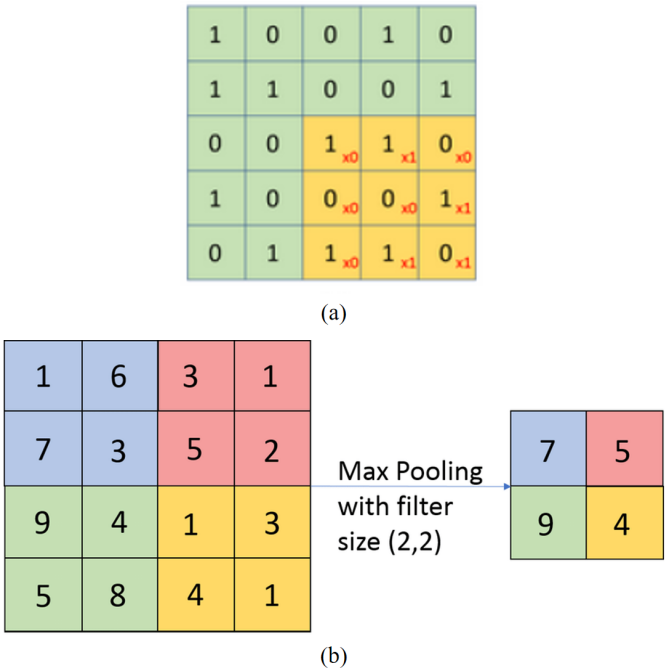


Figure 4. (a) Convolutional operation with a feature kernel followed by max pooling; (b) Max pooling with a filter size of (2, 2), selecting input from the feature kernel with similar positioning.

3.2.3 *Output Layer*

The output layer produces the network's final prediction or inference from the features learned by the preceding layers [13]. It employs an activation function, with Softmax being the most common for classification problems. This activation function scales the probability distribution over the output vector and returns it, with the most probable outcome selected as the predicted category.

3.2.4 *Activation Functions*

Activation functions define the output of a neuron and perform operations on the input, determining whether a neuron fires. They also introduce non-linearity between input and output, which is

essential for learning complex patterns. Different activation functions are selected based on the desired outcome [50]. In our proposed model, we employ ReLU and Softmax functions.

ReLU Activation Function The Rectified Linear Unit (ReLU) is a widely adopted activation function in image classification tasks. The ReLU function yields two possible outcomes: for a negative input, the output is zero; for a positive input, the output equals the input value [51, 52]. Formally, the ReLU activation function is defined as:

$$\mathrm{ReLU}(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{1}$$

This non-linear activation introduces sparsity and mitigates the vanishing gradient problem, thereby facilitating more efficient training of deep networks.

Softmax Activation Function The Softmax activation function is commonly employed in the output layer of neural networks for multi-class classification problems. It transforms a vector of real-valued logits into a probability distribution over multiple classes [52]. The Softmax function is mathematically defined as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad i = 1, 2, \ldots, K \tag{2}$$

where z_i denotes the input logit for the i-th class, e is the base of the natural logarithm, and K is the total number of classes. The Softmax operation exponentiates each logit to ensure positivity and subsequently normalizes each exponentiated value by the sum over all classes, guaranteeing that the output values lie within $[0, 1]$ and sum to unity, thus conforming to a valid probability distribution.

3.3 Implementation Framework

After selecting development tools, libraries, and packages, we proceeded to the architecture design and coding phase. All work was performed in the Anaconda Navigator GUI using Spyder (Python 3.8 environment). Spyder is an integrated development environment (IDE) specifically designed for scientific computing and data analysis with Python. It provides a user-friendly interface for exploring, analyzing, and visualizing data, as well as writing and running Python code. Spyder's features include an interactive console, a variable explorer, a debugger, a profiler, and a code editor with syntax highlighting and auto-completion, making it a popular choice for deep learning and neural network projects.

3.4 Dataset Description

Given the specialized nature of fruit quality classification applications, existing datasets are limited. We obtained a fresh-and-rotten fruit image dataset from Kaggle and also collected secondary data to supplement our experiments [53]; a related public fruit-image dataset is Fruits-360 [54]. The final dataset comprises 13,568 images of fresh and rotten fruits, including apples, bananas, and oranges. Each image was manually annotated by domain experts into six categories: fresh apples, fresh bananas, fresh oranges, rotten apples, rotten bananas, and rotten oranges.

Images were collected from publicly available repositories and supplemented with additional photographs captured under varying real-world conditions to ensure diversity in lighting, orientation, background, and fruit size. Prior to model training, all images were resized to 64×64 pixels and normalized to a [0,1] range. The dataset was partitioned into training (10,880 images, 80%) and testing (2,688 images, 20%) subsets, ensuring representative samples of each class. Table 1 provides a detailed overview of the dataset distribution, while Figure 5 shows example images from each class.

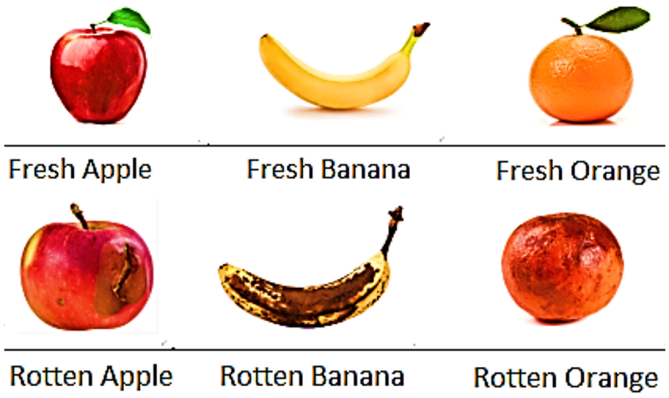


Figure 5. Fruit classification datasets, showing example images of fresh and rotten fruits used to train and evaluate the CNN model for accurate quality assessment.

Despite these efforts, potential dataset biases remain, such as slightly uneven class distribution (rotten fruit images outnumbering fresh ones) and underrepresented environmental conditions. These limitations were addressed through data augmentation and stratified sampling to reduce bias and improve evaluation reliability. Overall, this dataset provides a comprehensive and diverse set of

Table 1. Overview of the fruit classification dataset, listing the number of images for fresh and rotten fruits used to train and evaluate the CNN model.

Class	Total Images	Training Images	Testing Images
Fresh Apples	2,080	1,690	390
Fresh Bananas	1,960	1,580	380
Fresh Oranges	1,848	1,460	388
Rotten Apples	2,940	2,340	600
Rotten Bananas	2,750	2,220	530
Rotten Oranges	1,990	1,590	400
Total	**13,568**	**10,880**	**2,688**

fruit images suitable for developing and evaluating classification models for real-world fruit quality assessment.

3.5 Data Augmentation

Data augmentation is a commonly used technique in deep learning to increase the size of a training dataset by generating new versions of existing images. By applying domain-specific transformations such as rotations, reflections, and scaling to existing training examples, we can create new examples that remain representative of the original class. This improves the generalization ability of deep learning models, leading to better performance on unseen data.

Keras provides a convenient implementation of data augmentation through the `ImageDataGenerator` class, which allows users to specify transformations to be applied during training. It is important to note that these transformations should not be applied to validation or test datasets, as this would lead to overfitting and inaccurate evaluation. The selection of appropriate transformations depends on the dataset and problem domain; different datasets may require different transformations [43, 55]. Figure 6 shows a single augmented apple image using `ImageDataGenerator`.

The data augmentation process combines traditional and custom augmentations, ensuring diverse and realistic training data. The augmented image is obtained as:

$$x' = Preprocess(Augment(x)) \tag{3}$$

where Augment applies:

$$Rotation(\theta), \quad \theta \in [-40°, 40°] \tag{4}$$

$$Shift(x, \Delta_x, \Delta_y), \quad \Delta_x, \Delta_y \in [-20\%, 20\%] \tag{5}$$

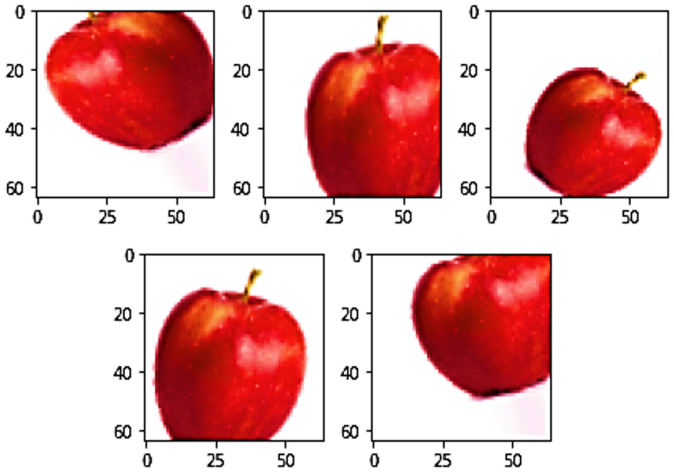


Figure 6. Example of a single apple image after augmentation using Keras ImageDataGenerator, demonstrating transformations such as rotation, zoom, and flipping applied to increase training data diversity.

$$Zoom(s), \quad s \in [80\%, 120\%] \tag{6}$$

$$Brightness(x, \beta), \quad \beta \in [0.8, 1.2] \tag{7}$$

$$ChannelShift(x, \Delta_c), \quad \Delta_c \in [-30, 30] \tag{8}$$

And preprocessing applies:

$$x_{crop} = x[y : y + h, x : x + w], \quad h, w = 56 \tag{9}$$

$$x' = Resize(x_{crop}, 64 \times 64) \tag{10}$$

3.6 Training Monitoring and ModelCheckpoint Callback

Training deep learning models can be computationally intensive, often requiring several days to converge. Therefore, it is essential to implement mechanisms for monitoring training progress and intervening when necessary—for instance, to halt training in cases of overfitting or when the loss has stabilized. Moreover, deep learning models are prone to unexpected interruptions due to their complexity, which can render extended training efforts futile.

Keras provides a suite of callback functions for proactive handling of such issues during training. The `ModelCheckpoint` callback, importable from `keras.callbacks`, is particularly useful. It saves the model at the end of each epoch, allowing users to specify storage location and the metric to be monitored. A high-level overview of the Keras `ModelCheckpoint` callback is illustrated in Figure 7.

3.7 Proposed CNN Model Architecture

The proposed model's strength lies in the hierarchical structure of its convolutional layers, combined

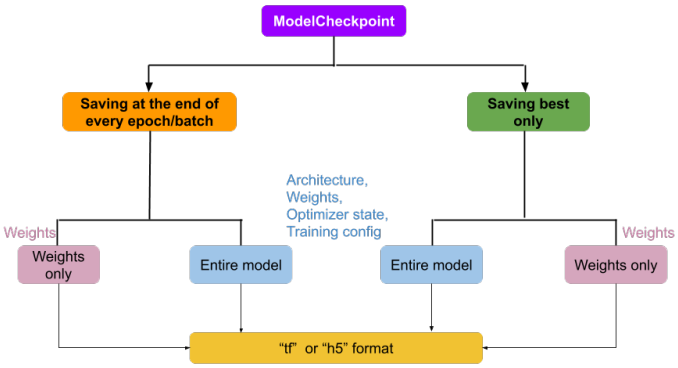


Figure 7. High-level overview of the Keras ModelCheckpoint callback, illustrating how model weights are automatically saved during training to preserve the best-performing CNN model.

with custom preprocessing and data augmentation techniques. Figure 8 shows the overall project implementation of the CNN model and web application, while Figure 9 illustrates the proposed CNN architecture and methodology flowchart.

Figure 10 depicts the fully connected layers, and Figure 11 displays the feature extraction and classification process of the proposed model.

The architecture can be expressed as:

$$\text{CNN Output} = \text{Softmax}\left(\text{Dense}\left(f\left(\text{flatten}(g(x))\right)\right)\right) \tag{11}$$

where x is the input image of dimensions $64 \times 64 \times 3$, $g(x)$ represents the feature extraction layers:

$$g(x) = \text{Conv4}\left(\text{Conv3}\left(\text{Conv2}\left(\text{Conv1}(x)\right)\right)\right) \tag{12}$$

Each Conv layer applies:

$$\text{ConvN}(z) = \text{MaxPooling}\left(\text{ReLU}\left(\text{Conv2D}(z, W_N)\right)\right) \tag{13}$$

where z is the input to the convolutional layer, W_N are the weights learned during training for the N-th convolutional layer, and MaxPooling applies downsampling to reduce spatial dimensions.

The fully connected layers are represented as:

$$f(x) = \text{Dense5}(\text{Dense4}(\text{Dense3}(\text{Dense2}(\text{Dense1}(x))))) \tag{14}$$

The final Dense layer outputs probabilities for 6 classes using the Softmax function:

$$\text{Softmax}(z) = \frac{e^{z_j}}{\sum_{j=1}^{C} e^{z_j}}, \quad i = 1, \ldots, C \tag{15}$$

A summary of the sequential CNN model architecture is provided in Table 2.

3.8 Loss Function and Optimization

3.8.1 Categorical Cross-Entropy Loss

For a given input image x, the model predicts a probability distribution $\hat{y}$ over C classes using the Softmax activation function. The true label is

Table 2. Summary of the sequential CNN model architecture, detailing layer types, output shapes, and parameters.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 128)	0
conv2d_3 (Conv2D)	(None, 4, 4, 256)	295,168
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 256)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 512)	524,800
dense_1 (Dense)	(None, 256)	131,328
dense_2 (Dense)	(None, 128)	32,896
dense_3 (Dense)	(None, 64)	8,256
dense_4 (Dense)	(None, 32)	2,080
dense_5 (Dense)	(None, 6)	198
	Total params	**1,087,974**
	Trainable params	1,087,974
	Non-trainable params	0

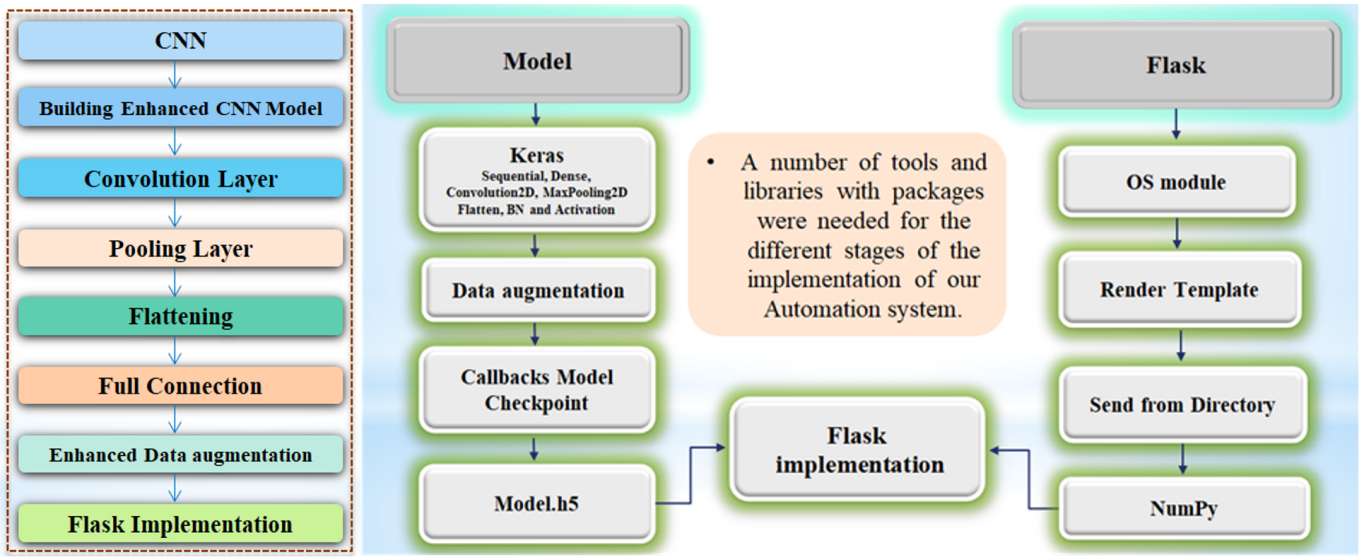


Figure 8. (a) Architecture of the implemented CNN for fruit classification, and (b) Deployed Flask application model showing integration for real-time fruit quality prediction.

```
# Initializing CNN
classifier = Sequential()

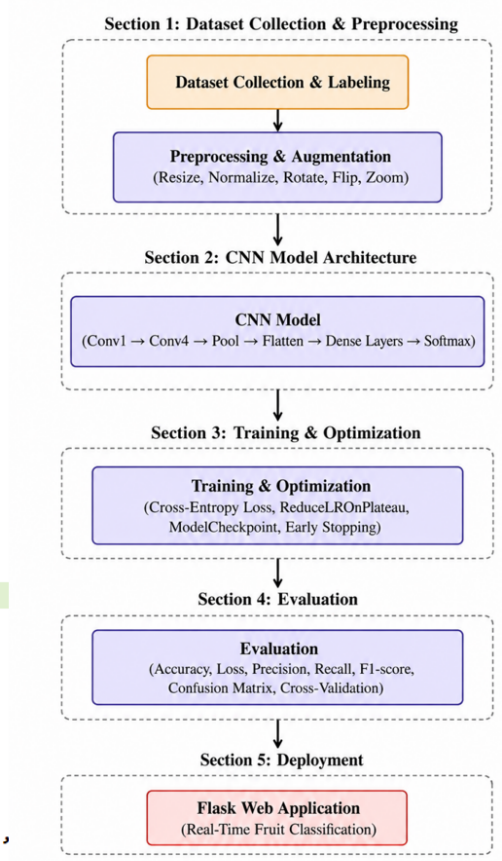
# Adding convolutional and pooling layers
# First Convolution Layer
classifier.add(Convolution2D(32, (3, 3), input_shape=(64, 64, 3),
                             activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Second Convolution Layer
classifier.add(Convolution2D(64, (3, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Third Convolution Layer
classifier.add(Convolution2D(128, (3, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Fourth Convolution Layer
classifier.add(Convolution2D(256, (3, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

#    Flattening the layers
classifier.add(Flatten())
# Fully Connected Dense Layers
classifier.add(Dense(units=512, activation='relu'))
classifier.add(Dense(units=256, activation='relu'))
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=64, activation='relu'))
classifier.add(Dense(units=32, activation='relu'))
classifier.add(Dense(units=6, activation='softmax'))
# Compiling the CNN
classifier.compile(optimizer='adam', loss='categorical_crossentropy',
                   metrics=['accuracy'])
```



(a) (b)

Figure 9. (a) Proposed CNN model for fruit classification, illustrating the sequence of convolutional, pooling, and fully connected layers; (b) Flowchart of proposed methodology.

represented as a one-hot encoded vector y. The cross-entropy loss is defined as:

$$L_{CCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \tag{16}$$

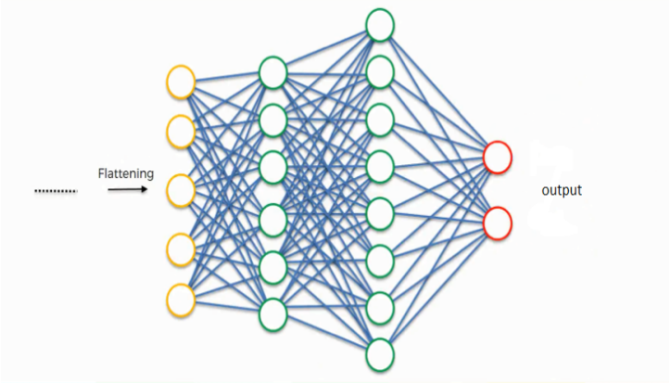


Figure 10. Fully connected layers in the CNN, showing how extracted features are flattened and passed through dense layers for final classification.

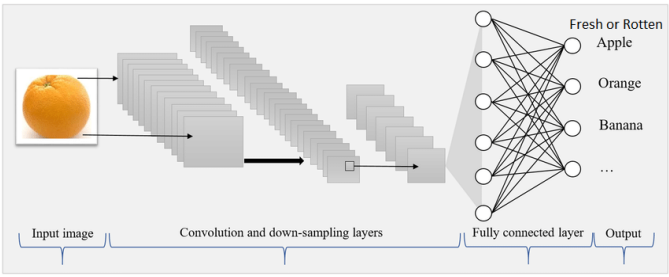


Figure 11. Feature extraction and classification process in the proposed CNN model, illustrating how learned features are used to classify fresh and rotten fruits.

where N is the number of training samples in the batch, C is the number of classes ($C = 6$), y_{ij} is the true label for sample i in class j, and $\hat{y}_{ij}$ is the predicted probability:

$$\hat{y}_{ij} = \frac{e^{z_{ij}}}{\sum_{k=1}^{C} e^{z_{ik}}} \tag{17}$$

where z_{ij} is the logit (raw output) of the network for sample i and class j.

The total loss is calculated as the average categorical cross-entropy loss over all samples in the batch:

$$L_{total} = \frac{1}{B} \sum_{i=1}^{B} L_{CCE}(x_i, y_i) \tag{18}$$

where B is the batch size.

3.8.2 Learning Rate Adjustment with ReduceLROnPlateau

To optimize the training process, the learning rate is adaptively adjusted using the ReduceLROnPlateau scheduler. The learning rate η is reduced when the validation loss L_{val} stagnates for a specified number of epochs:

$$\eta_{new} = \eta_{current} \times factor \quad \text{if } L_{val} \text{ does not improve for } p \text{ epochs} \tag{19}$$

where $factor$ is the reduction factor (e.g., 0.5), p is the number of epochs to wait before reducing the learning rate, and η_{min} is the minimum allowable learning rate.

3.9 Required Modules and Libraries

The main step before implementing the CNN model is importing the necessary modules and libraries. The Keras library is widely used for deep learning modules, and we utilize it extensively in this work. First, we import the sequential model from `keras.model`, which provides two classes: Sequential and Functional. We use the Sequential class, as it allows adding layers step by step, with the output of each layer serving as input to the next. Then, we import the required layers: Conv2D, Flatten, Dense, and MaxPooling2D. The Dense layer represents the artificial neural network, and MaxPooling2D is used for downsampling (see Figure 12).

```
#Importing keras libraries and packages

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ReduceLROnPlateau
import tensorflow as tf
import numpy as np
#for accuracy and loss graph
import matplotlib.pyplot as plt
```

Figure 12. Required modules and libraries for the CNN implementation, highlighting the software components used for building, training, and deploying the fruit classification model.

Next, we import `ModelCheckpoint` from `keras.callbacks` to save the best model based on validation accuracy. The `ImageDataGenerator` from `keras.preprocessing.image` is imported for data augmentation, enabling the conversion of a single image into multiple variations to train the model on more diverse data. Finally, we import the matplotlib library for generating accuracy and loss graphs.

4 Results

This section presents the experimental results of the proposed Enhanced CNN model for fruit classification. We evaluate the model's performance through data augmentation analysis, training accuracy, classification metrics, cross-validation, and comparison with state-of-the-art methods.

4.1 Data Augmentation Results

Data augmentation is the process of generating new data points from existing data to artificially increase

the dataset size. By applying subtle adjustments to the data, we create variations that improve model generalization. Figure 13 displays augmented images generated from the training data using the `plotimages()` function, which arranges images in a grid with 1 row and 5 columns.

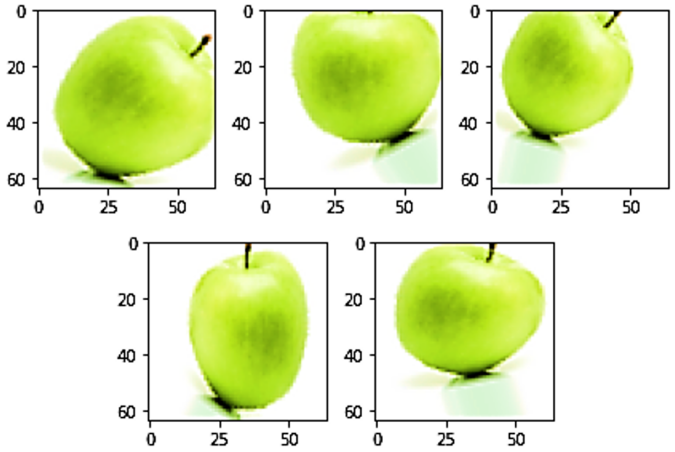


Figure 13. Example results of image augmentation, showing how fruit images are transformed (e.g., rotated, flipped, zoomed) to enhance dataset diversity for CNN training.

4.2 Model Training and Checkpointing

To save the best-performing model during training, we employed the `ModelCheckpoint` callback from Keras. This callback monitors the validation accuracy throughout the training process and saves the model whenever an improvement is observed. During the first epoch, the training accuracy was 49.94% with a validation accuracy of 65.89%. The model continued to improve until epoch 56, achieving a training accuracy of 99.07% and a validation accuracy of 99.22%. Beyond epoch 56, no further improvement in validation accuracy was observed. However, for the final deployment, we utilized the `last_checkpoint.h5` model saved at epoch 300, as it demonstrated superior performance in practical fruit classification and prediction tasks.

4.3 Accuracy of the Final Implemented Model

The enhanced CNN model, saved as `last_checkpoint.h5` at epoch 300, was selected for the final Flask web application implementation due to its superior classification performance. Throughout the training process, both training accuracy and validation accuracy gradually increased, while training loss and validation loss consistently decreased. At the final epoch, the model achieved a training accuracy of 99.46% and a validation accuracy of 99.03%, demonstrating effective learning and generalization.

4.4 Experimental Setup and Results

The system was compiled and trained using the Fruit dataset, comprising 13,568 images across six categories: fresh apples, fresh bananas, fresh oranges, rotten apples, rotten bananas, and rotten oranges. The dataset was split into 80% training (10,880 images) and 20% testing (2,688 images). The model was trained for 300 epochs with a batch size of 32. The average training time per epoch was approximately 70 seconds on an AMD RYZEN 7 (8 cores), which is reasonable given the dataset size and network complexity. The proposed model achieved a training accuracy of 99.46% and a validation accuracy of 99.03%. Figure 14 illustrates the classification and prediction results.

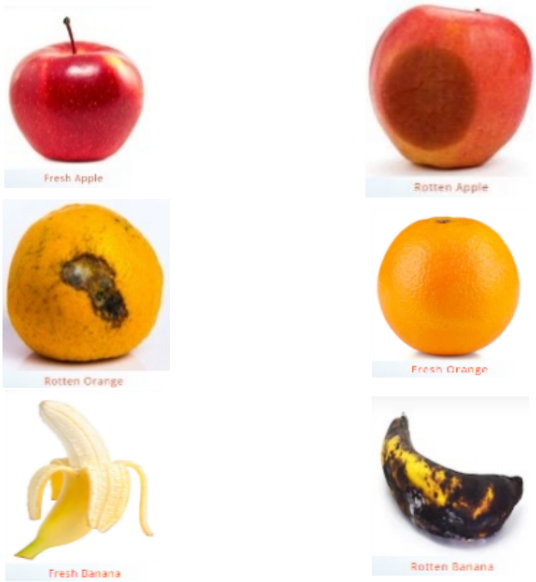


Figure 14. Performance results of the proposed CNN model, illustrating classification accuracy and effectiveness in distinguishing fresh and rotten fruits.

4.5 Classification Report

Table 3 presents a comprehensive classification report for the proposed Enhanced CNN model, detailing precision, recall, F1-score, and support for each of the six fruit categories.

4.6 Confusion Matrix Analysis

The confusion matrix presented in Figure 15 illustrates the classification performance across all six fruit categories. While the proposed Enhanced CNN model achieved high overall accuracy, certain misclassification trends were observed. The diagonal

Table 3. Classification report of the proposed CNN model, showing precision, recall, F1-score, and support for each fruit category.

Class	Precision	Recall	F1-Score	Support
Fresh Apples	0.99	0.98	0.99	390
Fresh Bananas	1.00	1.00	1.00	380
Fresh Oranges	1.00	0.97	0.98	388
Rotten Apples	0.97	1.00	0.98	600
Rotten Bananas	1.00	1.00	1.00	530
Rotten Oranges	0.99	0.98	0.99	400
Macro Avg	0.99	0.99	0.99	2,688
Weighted Avg	0.99	0.99	0.99	2,688

elements represent correct classifications, while off-diagonal elements indicate misclassifications.

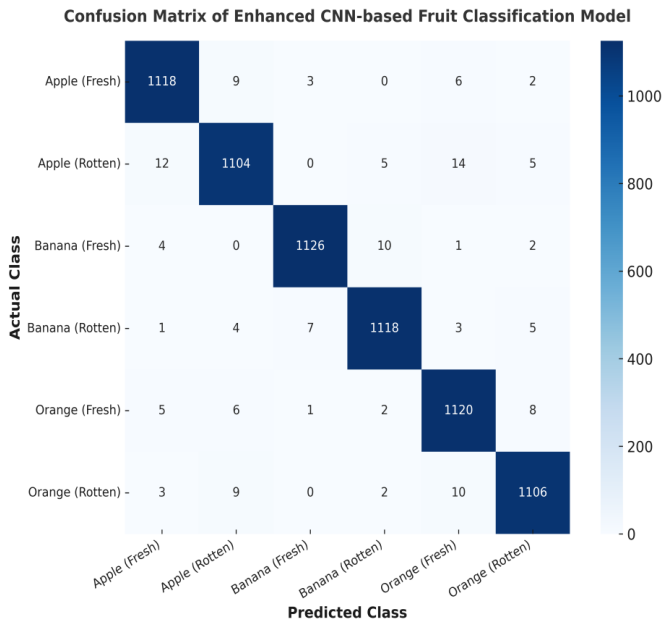


Figure 15. Confusion matrix of the proposed CNN model, showing the number of correct and incorrect predictions for each fruit category to evaluate classification performance.

The analysis reveals that the model occasionally confuses visually similar categories, particularly between rotten apples and rotten oranges, due to their color and texture similarity under variable lighting conditions. These errors highlight the influence of illumination, occlusion, and dataset imbalance. Although data augmentation helped mitigate these issues, complete elimination remains challenging. The model's accuracy slightly decreases under uncontrolled lighting and complex backgrounds, suggesting that real-world deployment may require additional fine-tuning and larger, more diverse datasets. Despite these limitations, the proposed model maintains robust classification performance. Figure 15 presents the confusion matrix of the proposed model.

4.7 Accuracy and Loss Curves

To monitor training progress, we utilized Keras History objects and Matplotlib for visualization. The `classifier.fit_generator` function returns a History object containing training and validation accuracy and loss values for each epoch. We plotted both metrics using Matplotlib's subplot feature to display training/validation accuracy and training/validation loss in a single window, as shown in Figures 16 and 17.

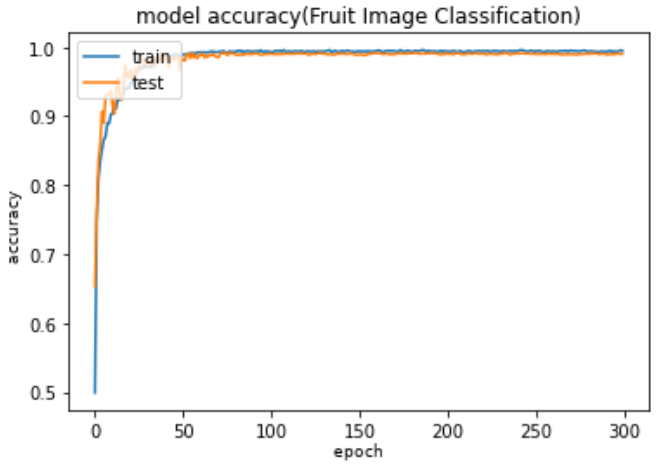


Figure 16. Accuracy graph of the proposed CNN model, showing how classification accuracy evolves over training epochs.

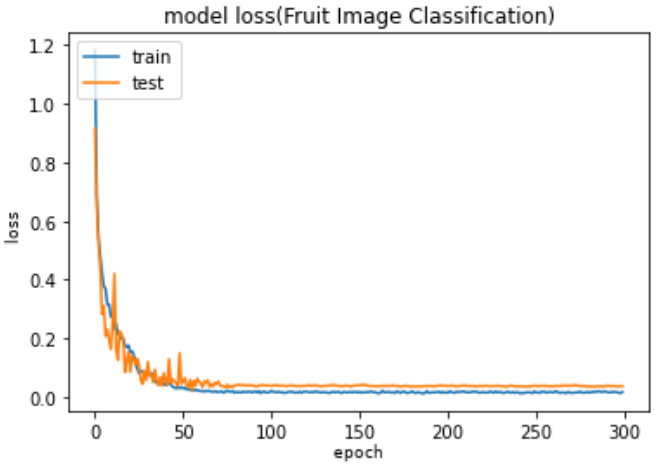


Figure 17. Loss graph of the proposed CNN model, illustrating the reduction in training and validation loss over epochs, indicating model convergence.

The accuracy and loss curves show that the model converges after roughly 75 epochs. After convergence the training accuracy is slightly higher and the training loss lower than the corresponding validation values (validation loss of about 0.04 against about 0.015 for training), which indicates a small generalization gap. The ReduceLROnPlateau scheduler helped to stabilize training.

4.8 Cross-Validation

Table 4. Cross-validation results across folds, presenting the classification accuracy (%) of the proposed model for each fold.

Fold	Accuracy (%)
1	99.40
2	99.45
3	99.50
4	99.38
5	99.42
Mean ± Std	**99.43 ± 0.05**

To provide a statistically robust evaluation, we applied 5-fold stratified cross-validation to the training set (10,880 images). Each fold preserves the proportion of classes to ensure representative sampling. Table 4 presents the classification accuracy achieved by the proposed model for each fold.

The low standard deviation (0.05%) demonstrates consistent performance across all folds, indicating that the model is stable and generalizes well to different subsets of the dataset.

4.9 Confidence Intervals

Using the cross-validation results, a 95% confidence interval (CI) for accuracy was computed using Eq. 20:

$$CI = \bar{x} \pm \frac{t_{\alpha/2,n-1} \cdot s}{\sqrt{n}} \tag{20}$$

where $\bar{x} = 99.43\%$ is the mean accuracy, $s = 0.047\%$ is the standard deviation of the folds, $n = 5$ is the number of folds, and $t_{0.025,4} \approx 2.776$ is the t-value for 95% CI with 4 degrees of freedom. Substituting these values gives:

$$CI = 99.43 \pm 2.776 \cdot \frac{0.047}{\sqrt{5}} = 99.43 \pm 0.058 \tag{21}$$

$$\text{95\% CI: } 99.37\% - 99.49\% \tag{22}$$

This narrow confidence interval confirms that the model's accuracy is highly reliable and unlikely to vary significantly on unseen data.

4.10 Comparison with State-of-the-Art Methods

The proposed Enhanced CNN model was compared against existing publicly reported methods using similar fruit classification datasets. Table 5 presents a comprehensive comparison of various techniques, their datasets, and evaluation metrics.

The proposed model achieves a validation accuracy of 99.03% (training accuracy 99.46%), which is higher than Han et al. [35] (96.1%), Hossain et al. [36] (97.5%), Femling et al. [56] (97% top-3 and 76% top-1 accuracy), and the best CNN of Taner et al. [34] (97.48%). The deep-feature pipeline of Taner et al. [34] (DenseNet201 features, PCA and MLP) reports 99.77% on a different dataset of apple varieties, and Gulzar [33] and Sangeetha and Raja [31] report 99.00%, so the proposed model is comparable to the best reported results rather than superior to all of them. The 5-fold cross-validation results (mean: 99.43% ± 0.05%) and the 95% confidence interval (99.37%–99.49%) further confirm the model's robustness and generalization capability. Given the differences in datasets and tasks, the comparison is indicative only, but it shows that the Enhanced CNN is competitive for reliable fruit classification under variable conditions.

4.11 Discussion

The proposed Enhanced CNN model with data augmentation achieved a validation accuracy of 99.03% (training accuracy 99.46%), which is higher than most of the compared methods and comparable to the best reported values; differences in datasets and tasks mean that the comparison is indicative only. The 5-fold cross-validation results (mean accuracy: 99.43% ± 0.05%) and the narrow 95% confidence interval (99.37% – 99.49%) confirm the model's robustness and generalization capability.

Compared to state-of-the-art methods, the proposed model demonstrates several advantages. First, it handles a larger and more diverse dataset (13,568 images across 6 classes) while maintaining higher accuracy than methods using smaller datasets. Second, it effectively distinguishes between visually similar fruits and between fresh and rotten categories. Third,

Table 5. Comparative analysis of existing and proposed fruit classification techniques.

Author	Fruit Type / Dataset Size / Classes	Technique / Model Type	Evaluation Metrics
Han et al. [35]	Pear dataset, 1,800 images, 2 classes	ANN	Accuracy (96.1%), Precision, Recall
Hossain et al. [36]	15 fruit categories, 5,946 images	VGG-16 (Pre-trained CNN)	Accuracy (97.5%), F1-Score
Femling et al. [56]	10 fruit/vegetable classes, 4,300 images	Inception, MobileNet (retrained CNNs)	Top-3 accuracy (97% MobileNet, 96% Inception), top-1 accuracy (76%), Propagation Time
Kathepuri [57]	Fruits-360, 131 classes, 90,483 images	Base CNN	Accuracy (96.91%), Training Time
Bird et al. [55]	Lemons, 2,690 images, 2 classes	VGG16 + CGAN Data Augmentation	Accuracy (88.75%), Precision
Sangeetha and Raja [31]	Mixed fruit, fresh vs. rotten	CNN + OpenCV	Accuracy (99.00%), F1-Score, Sensitivity
Wang et al. [32]	33 fruit/veg categories, 22,495 images	SVM	Accuracy (96.25%), Recall
Gulzar [33]	40 fruit categories, 26,149 images	MobileNetV2 (transfer learning)	Accuracy (99.00%), Equal Error Rate
Taner et al. [34]	13 apple varieties, 5,000 images	VGG16, VGG19, InceptionV3, MobileNet, Xception, ResNet150V2, DenseNet201 (best CNN); DenseNet201 deep features + PCA + MLP	Accuracy (97.48% CNN; 99.77% with PCA + MLP), Precision, Recall
Proposed Model	Apple, Banana, Orange, 13,568 images, 6 classes	Enhanced CNN + Data Augmentation	**Accuracy (99.03%, validation), Precision (99%), Recall (99%), F1-Score (99%)**

the comprehensive evaluation using multiple metrics (accuracy, precision, recall, F1-score) provides a thorough assessment of model performance.

The confusion matrix analysis revealed minor misclassifications between visually similar categories, particularly under challenging conditions such as uncontrolled lighting and complex backgrounds. These limitations suggest that future work should focus on expanding the dataset with more diverse samples and incorporating additional preprocessing techniques.

The prototype Flask web application demonstrates the practical applicability of the proposed model for real-world fruit quality assessment. If commercialized, such a system could significantly reduce human error, improve operational efficiency, and enhance profitability for distributors and retailers in the agricultural industry.

5 Conclusion

This paper presents an innovative approach for fruit image classification using an Enhanced Convolutional Neural Network (CNN) with data augmentation to distinguish between fresh and rotten apples, bananas, and oranges. The proposed model was implemented in Python 3.8 using the Anaconda Spyder IDE and deployed as a Flask-based web application for real-time fruit quality assessment. The dataset comprised 13,568 images across six categories, partitioned into training (80%) and testing (20%) subsets. The model was trained for 300 epochs with a batch size of 32, achieving a training accuracy of 99.46% and a validation accuracy of 99.03%.

To ensure statistical robustness, 5-fold stratified cross-validation was applied, yielding a mean accuracy of 99.43% $\pm$ 0.05% with a 95% confidence interval of 99.37% – 99.49%, demonstrating the model's stability and generalization capability. Comprehensive evaluation using precision, recall, F1-score, and confusion matrix analysis confirmed the model's effectiveness in accurately classifying fresh and rotten fruits. Comparative analysis against state-of-the-art methods, including VGG-16, MobileNet, and hybrid CNN architectures, showed that the proposed model achieves accuracy comparable to or higher than the reported results of existing approaches.

The successful development and deployment of this system highlight the significant potential of deep learning in agricultural applications. Such technology can enhance food quality assurance, reduce manual inspection errors, and improve supply chain efficiency for distributors and retailers. The findings of this

research contribute to the growing body of knowledge on CNN-based fruit classification and provide a foundation for future advancements in automated quality assessment systems.

5.1 Future Work

Although the proposed CNN model demonstrates high accuracy in classifying fresh and rotten fruits, several avenues remain for future improvement:

- **Expansion to Larger and Diverse Datasets**: Extending the dataset to include more fruit varieties, different ripening stages, and varied environmental conditions (lighting, occlusion, backgrounds) could improve model robustness and applicability in real-world scenarios.
- **Integration with Edge Devices**: Deploying the model on portable edge devices or smartphones for real-time fruit quality assessment in farms and markets could enhance practical usability. Techniques such as model quantization and pruning can help reduce computational cost and latency.
- **Multi-Modal Inputs**: Incorporating additional sensory data, such as weight, color spectrum analysis, or hyperspectral imaging, could improve classification accuracy, especially for visually similar fruits.
- **Explainable AI (XAI)**: Implementing explainable deep learning techniques could provide insights into the decision-making process of the CNN, aiding trust and interpretability for agricultural practitioners.
- **Hybrid and Ensemble Models**: Combining CNN with other machine learning or deep learning architectures (e.g., attention mechanisms, Transformers) could further improve classification performance, particularly for challenging cases such as partially rotten or damaged fruits.
- **Automation in Smart Farming Pipelines**: Integrating the CNN with robotic sorting systems or IoT-based smart farming platforms can enable fully automated fruit grading, reducing manual labor and improving operational efficiency.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Khaing, Z. M., Naung, Y., & Htut, P. H. (2018, January). Development of control system for fruit classification based on convolutional neural network. In *2018 IEEE conference of russian young researchers in electrical and electronic engineering* (*EIConRus*) (pp. 1805-1807). IEEE. [CrossRef]

[2] Mukhiddinov, M., Muminov, A., & Cho, J. (2022). Improved classification approach for fruits and vegetables freshness based on deep learning. *Sensors, 22*(21), 8192. [CrossRef]

[3] Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R. J., Fredes, C., & Valenzuela, A. (2020). A review of convolutional neural network applied to fruit image processing. *Applied Sciences, 10*(10), 3443. [CrossRef]

[4] Zhang, Y., & Wu, L. (2012). Classification of fruits using computer vision and a multiclass support vector machine. *sensors, 12*(9), 12489-12505. [CrossRef]

[5] Wang, L., Li, A., & Tian, X. (2013, November). Detection of fruit skin defects using machine vision system. In *2013 Sixth International Conference on Business Intelligence and Financial Engineering* (pp. 44-48). IEEE. [CrossRef]

[6] Wu, A., Zhu, J., & Ren, T. (2020). Detection of apple defect using laser-induced light backscattering imaging and convolutional neural network. *Computers & Electrical Engineering, 81*, 106454. [CrossRef]

[7] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and electronics in agriculture, 147*, 70-90. [CrossRef]

[8] Amin, U., Shahzad, M. I., Shahzad, A., Shahzad, M., Khan, U., & Mahmood, Z. (2023). Automatic fruits freshness classification using CNN and transfer learning. *Applied Sciences, 13*(14), 8087. [CrossRef]

[9] Sahu, D., & Dewangan, C. (2017). Identification and classification of mango fruits using image processing. *International Journal of Scientific Research in Computer*

Science, Engineering and Information Technology, *2*(2), 203-210. https://ijsrcseit.com/CSEIT172271

[10] Dubey, S. R., & Jalal, A. S. (2015). Application of image processing in fruit and vegetable analysis: a review. *Journal of Intelligent Systems*, *24*(4), 405-424. [CrossRef]

[11] Alegbeleye, O. O., Singleton, I., & Sant'Ana, A. S. (2018). Sources and contamination routes of microbial pathogens to fresh produce during field cultivation: A review. *Food Microbiology*, *73*, 177-208. [CrossRef]

[12] Palakodati, S. S. S., Chirra, V. R. R., Yakobu, D., & Bulla, S. (2020). Fresh and Rotten Fruits Classification Using CNN and Transfer Learning. *Rev. d'Intelligence Artif.*, *34*(5), 617-622. [CrossRef]

[13] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444. [CrossRef]

[14] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, *2018*(1), 7068349. [CrossRef]

[15] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, *77*, 354-377. [CrossRef]

[16] Ha, J. G., Moon, H., Kwak, J. T., Hassan, S. I., Dang, M., Lee, O. N., & Park, H. Y. (2017). Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial vehicles. *Journal of Applied Remote Sensing*, *11*(4), 042621-042621. [CrossRef]

[17] Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z., & Sun, Z. (2018). A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and electronics in agriculture*, *154*, 18-24. [CrossRef]

[18] Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, *267*, 378-384. [CrossRef]

[19] Liu, B., Zhang, Y., He, D., & Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, *10*(1), 11. [CrossRef]

[20] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in plant science*, *7*, 1419. [CrossRef]

[21] Tran, T. T., Choi, J. W., Le, T. T. H., & Kim, J. W. (2019). A comparative study of deep CNN in forecasting and classifying the macronutrient deficiencies on development of tomato plant. *Applied Sciences*, *9*(8), 1601. [CrossRef]

[22] Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, *17*(9), 2022. [CrossRef]

[23] Wang, G., Sun, Y., & Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. *Computational intelligence and neuroscience*, *2017*(1), 2917536. [CrossRef]

[24] Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2017). Big data in smart farming – A review. *Agricultural Systems*, *153*, 69-80. [CrossRef]

[25] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and electronics in agriculture*, *157*, 417-426. [CrossRef]

[26] Bargoti, S., & Underwood, J. P. (2017). Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics*, *34*(6), 1039-1060. [CrossRef]

[27] Xue, G., Liu, S., & Ma, Y. (2023). A hybrid deep learning-based fruit classification using attention model and convolution autoencoder. *Complex & Intelligent Systems*, *9*(3), 2209-2219. [CrossRef]

[28] Rojas-Aranda, J. L., Nunez-Varela, J. I., Cuevas-Tello, J. C., & Rangel-Ramirez, G. (2020, June). Fruit classification for retail stores using deep learning. In *Mexican Conference on Pattern Recognition* (pp. 3-13). Cham: Springer International Publishing. [CrossRef]

[29] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM computing surveys (CSUR)*, *51*(5), 1-36. [CrossRef]

[30] Thi Phuong Chung, D., & Van Tai, D. (2019, October). A fruits recognition system based on a modern deep learning technique. In *Journal of physics: conference series* (Vol. 1327, No. 1, p. 012050). IOP Publishing. [CrossRef]

[31] Sangeetha, K., & Raja, P. V. (2024, March). Classification of fruits and its quality prediction using deep learning. In *2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (pp. 342-346). IEEE. [CrossRef]

[32] Wang, W., Zhu, A., Wei, H., & Yu, L. (2024). A novel method for vegetable and fruit classification based on using diffusion maps and machine learning. *Current Research in Food Science*, *8*, 100737. [CrossRef]

[33] Gulzar, Y. (2023). Fruit image classification model based on MobileNetV2 with deep transfer learning technique. *Sustainability*, *15*(3), 1906. [CrossRef]

[34] Taner, A., Mengstu, M. T., Selvi, K. Ç., Duran, H., Gür, İ., & Ungureanu, N. (2024). Apple varieties classification using deep features and machine learning. *Agriculture*, *14*(2), 252. [CrossRef]

[35] Han, Z., Liu, J., Zhao, Y., & Li, Y. (2012, May). Grading system of pear's appearance quality based on computer vision. In *2012 International Conference on Systems and Informatics (ICSAI2012)* (pp. 184-188). IEEE. [CrossRef]

[36] Hossain, M. S., Al-Hammadi, M., & Muhammad, G. (2019). Automatic fruit classification using deep learning for industrial applications. *IEEE transactions on industrial informatics, 15*(2), 1027-1034. [CrossRef]

[37] Parfitt, J., Barthel, M., & Macnaughton, S. (2010). Food waste within food supply chains: quantification and potential for change to 2050. *Philosophical Transactions of the Royal Society B: Biological Sciences, 365*(1554), 3065-3081. [CrossRef]

[38] Godfray, H. C. J., Beddington, J. R., Crute, I. R., Haddad, L., Lawrence, D., Muir, J. F., Pretty, J., Robinson, S., Thomas, S. M., & Toulmin, C. (2010). Food security: The challenge of feeding 9 billion people. *Science, 327*(5967), 812-818. [CrossRef]

[39] Fountas, S., Mylonas, N., Malounas, I., Rodias, E., Hellmann Santos, C., & Pekkeriet, E. (2020). Agricultural robotics for field operations. *Sensors, 20*(9), 2672. [CrossRef]

[40] Karakaya, D., Ulucan, O., & Turkan, M. (2019, October). A comparative analysis on fruit freshness classification. In *2019 innovations in intelligent systems and applications conference (ASYU)* (pp. 1-4). IEEE. [CrossRef]

[41] Mureşan, H., & Oltean, M. (2018). Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica, 10*(1), 26-42. [CrossRef]

[42] Tapia-Mendez, E., Cruz-Albarran, I. A., Tovar-Arriaga, S., & Morales-Hernandez, L. A. (2023). Deep learning-based method for classification and ripeness assessment of fruits and vegetables. *Applied Sciences, 13*(22), 12504. [CrossRef]

[43] Zhang, Y. D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K., & Wang, S. H. (2019). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications, 78*(3), 3613-3632. [CrossRef]

[44] Hou, L., Wu, Q., Sun, Q., Yang, H., & Li, P. (2016, August). Fruit recognition based on convolution neural network. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (pp. 18-22). IEEE. [CrossRef]

[45] Lu, S., Lu, Z., Aok, S., & Graham, L. (2018, November). Fruit classification based on six layer convolutional neural network. In *2018 IEEE 23rd International conference on digital signal processing (DSP)* (pp. 1-5). IEEE. [CrossRef]

[46] Bhargava, A., & Bansal, A. (2021). Fruits and vegetables quality evaluation using computer vision: A review. *Journal of King Saud University Computer and Information Sciences, 33*(3), 243-257. [CrossRef]

[47] Marimuthu, S., & Roomi, S. M. M. (2017). Particle swarm optimized fuzzy model for the classification of banana ripeness. *IEEE Sensors Journal, 17*(15), 4903-4915. [CrossRef]

[48] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research, 15*(1), 1929-1958. https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf

[49] Sakib, S., Ashrafi, Z., & Siddique, M. A. B. (2019). Implementation of fruits recognition classifier using convolutional neural network algorithm for observation of accuracies for various hidden layers. *arXiv preprint arXiv:1904.00783*. [CrossRef]

[50] Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*. [CrossRef]

[51] Szandała, T. (2020). Review and comparison of commonly used activation functions for deep neural networks. In *Bio-inspired neurocomputing* (pp. 203-224). Singapore: Springer Singapore. [CrossRef]

[52] Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology, 4*(12), 310-316. [CrossRef]

[53] Kalluri, S. R. (2018). *Fruits fresh and rotten for classification* [Data set]. Kaggle. Retrieved from https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification

[54] Oltean, M. (2017). *Fruits-360 dataset* [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/moltean/fruits

[55] Bird, J. J., Barnes, C. M., Manso, L. J., Ekárt, A., & Faria, D. R. (2022). Fruit quality and defect image classification with conditional GAN data augmentation. *Scientia horticulturae, 293*, 110684. [CrossRef]

[56] Femling, F., Olsson, A., & Alonso-Fernandez, F. (2018, November). Fruit and vegetable identification using machine learning for retail applications. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 9-15). IEEE. [CrossRef]

[57] Kathepuri, S. (2020). *Recognition and classification of fruits using deep learning techniques* [Doctoral dissertation, National College of Ireland]. https://norma.ncirl.ie/4448/1/shubhamkathepuri.pdf



Muhammad Idrees received his Bachelor Degree in Computer Science from Hazara University, Pakistan in 2015 & Master Degree in Computer Science and Technology from The North China Electric Power University Beijing, China in 2023, respectively. He has more than 8 years of teaching experience. Currently, he is a PhD Scholar in School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China. He has serving as Teacher in Education department kpk, Pakistan. (Email: L202310006@stu.cqupt.edu.cn, idreeskhan045@gmail.com)



Altaf Hussain received his Bachelor Degree in Computer Science from University of Peshawar, Pakistan in 2013 & Master Degree in Computer Science from The University of Agriculture Peshawar, Pakistan in 2017, respectively. He has more than 6 years of teaching & research experience. He worked at The University of Agriculture Peshawar in Faculty of IT as Researcher from 2017 to 2019. He has supervised many bachelor's and master's degree level students and helped them with their final year projects and research. During his Master study, he has completed his research in drone communication systems. Currently, he is a PhD Scholar in School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China, specializing in Underwater Acoustic Sensor Networks with focus on energy efficiency, channel optimization, localization and dynamic scheduling integrating 6G and quantum computing. He has served as a Lecturer in Computer Science Department in Government Degree College Lal Qilla Dir Lower, KPK Pakistan from 2020 to 2021. He has worked as Research Assistant with the Department of Accounting and Information Systems, College of Business and Economics, Qatar University, Doha, Qatar. He also worked as IT clerk in the Court of District and Session Judge Timergara Dir Lower from 2022 to 2023. He has published several notable research papers. He has reviewed many articles and is serving as reviewer for Cluster Computing, Computing, Cybernetics and Systems, Journal of Cloud Computing, Knowledge and Information Systems, Peer-to-Peer Networking and Applications, SN Applied Sciences, The Imaging Science Journal, The Journal of Supercomputing, Transactions on Emerging Telecommunications Technologies, Wireless Personal Communications, Frontiers in Big Data, CMC-Computers, Materials & Continua, and Bulletin of Electrical Engineering and Informatics (BEEI). His Research interest includes Artificial Intelligence, Machine Learning, Deep Learning, Gesture Detection, Wireless Networks, Underwater Sensor Networks, and Unmanned Drone Communication Systems. (Email: email@email.com)



Nasir Hussain is a PhD candidate at Chongqing University of Posts and Telecommunications, specializing in IIoT security with a focus on cybersecurity Intrusion Detection. He earned his Master's in Computer Science from NUAA China. His broader research interests include IIoT security, artificial intelligence, cloud computing, GIS, Machine learning, and reinforcement learning. (Email: 202310005@stu.cqupt.edu.cn)



Usman J Wushishiis a PhD candidate at Chongqing University of Posts and Telecommunications, specializing in IIoT security with a focus on cybersecurity deception. He earned his Master's in Management Information Systems from CIU Nicosia. His broader research interests include artificial intelligence, deep learning, game theory, and reinforcement learning. (Email: l202310011@stu.cqupt.edu.cn)



Muhammad Imran received the B.S. degree in Information Technology from the University of Education, Pakistan, in 2018, and the M.S. degree in Computer Science from the University of Okara, Pakistan, in 2022. He is currently pursuing the Ph.D. degree in Computer Science at the University of Posts and Telecommunications, China. His research during the B.S. focused on face recognition, while his M.S. research was in the field of image processing. He is currently conducting research in Explainable Artificial Intelligence (XAI). In addition, he has undertaken research and coursework in networking and cybersecurity. (Email: L202310008@stu.cqupt.edu.cn)



Atif Ali Wagan is a is a PhD Scholar at School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China specializing in IIoT security with a focus on IDS. His broader research interests include artificial intelligence, computer vision, deep learning, game theory, and reinforcement learning. (Email: atif.wagan2@yahoo.com)