JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 10 Issue 01
2025

Research Article
https://doi.org/10.56979/1001/2025/1119

GAN-Augmented Deep Learning Model for Automated Fruit Ripeness Classification

Kanwal Saleem[1], Ayesha Hakim[1,2*], and Salman Qadri[1]

[1]Institute of Computing, MNS University of Agriculture, Multan, 60000, Pakistan.
[2]School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, 44000, Pakistan.
[*]Corresponding Author: Ayesha Hakim. Email: ayesha.hakim@seecs.edu.pk

Received: September 02, 2025 **Accepted:** November 10, 2025

Abstract: Manual fruit harvesting is costly, labor-intensive, and difficult to scale. We present a vision-based system for automated fruit-ripeness classification that jointly predicts fruit type and ripeness stage as an eight-class task {mango, strawberry, tomato, sweet pepper}×{ripe, unripe}. To mitigate limited and imbalanced field data, we train class-conditioned GANs to synthesize additional images and build an augmented training set (3,988 real images → 57,339 total). Synthetic samples are quality-controlled using automated filters and human review and are used only in training; validation and test splits contain real images exclusively. We benchmark three classifiers, DenseNet-201, ResNet-50, and a compact CNN, using a unified pipeline with standard preprocessing and on-the-fly augmentation. DenseNet-201 achieves the best generalization, reaching 99.41% training accuracy, 98.01% validation accuracy, and 95.7% accuracy on the held-out real-image test set, outperforming ResNet-50 and the CNN baseline on recall, precision, and F1 score. The results indicate that targeted generative augmentation improves robustness to variations in viewpoint, partial occlusion, and illumination, enabling reliable ripeness assessment from in-orchard imagery. The proposed pipeline provides a reproducible foundation for integrating ripeness classification into robotic harvesting workflows and can be extended to additional crops, sensing modalities, and on-device inference.

Keywords: Fruit Ripeness; Generative Adversarial Network; DenseNet-201; Data Augmentation; Transfer Learning; Reproducibility

1. Introduction

The agricultural industry has been facing a range of persistent challenges, including a declining number of farm laborers and rising costs associated with fruit harvesting. Manual harvesting methods are increasingly unsuitable for large-scale operations, as they are time-consuming, labor-intensive, and susceptible to inconsistencies. Yield estimation becomes particularly problematic due to leaf occlusion and the difficulty in distinguishing between different stages of fruit ripeness, which significantly increases both time and labor costs. Consequently, modern agricultural processes are progressively incorporating automation technologies, with computer vision, deep learning, and deep generative modeling replacing many traditional, labor-intensive tasks [1].

Fruit picking remains one of the most time-consuming and effort-intensive operations in the farming sector. Automation technologies that can reliably and efficiently identify fruits for harvesting have become essential. Deep learning offers a powerful framework for executing complex tasks such as object detection, classification, and real-time recognition, especially when integrated with machine vision systems. The development of fruit classification and recognition systems using deep learning algorithms has enabled the deployment of intelligent robotic harvesters capable of automating the picking process [2].

In the past few years, deep generative models have become an increasingly valuable approach for computer vision applications, especially those involving fruit recognition and determining ripeness levels [3]. Ripening is a natural phase in the development of fruits and vegetables, and its accurate identification holds significant value in agricultural automation. When paired with computer vision techniques, deep learning has demonstrated strong performance for problems of this nature [4].

The primary objective of this study is to enhance Automated Target Recognition (ATR) used in robotic fruit harvesting by incorporating deep generative modeling methods. Deep Generative Models (DGM) capture the fundamental structure of a dataset and create convincing synthetic images, helping address issues such as limited data availability and imbalanced classes in vision-related problems [5]. In agricultural imaging, Generative Adversarial Networks (GANs) can produce high-variation fruit images that reflect different viewpoints, lighting conditions, and occlusions, thereby improving the robustness of downstream classifiers [6]. In this study, we employed GANs to create a diverse, balanced dataset of ripe and unripe images for four target fruits, mango, strawberry, tomato, and sweet pepper, and use this GAN-augmented dataset to train and evaluate two transfer-learning classifiers (DenseNet-201 and ResNet-50). Performance comparisons are performed on both synthetic and real-farm images to identify the best model in terms of accuracy, inference speed, and real-world robustness [7].

By producing visually accurate and diverse synthetic images, DGM can significantly improve a vision system's ability to classify and recognize fruits under occlusion, shadow, or varying lighting conditions [6]. A typical use case involves the automated recognition of fruits such as mangoes using deep learning models [8]. In this work, we integrate GAN-based synthetic data generation with DenseNet-201 and ResNet-50 to build a complete Automated Target Recognition (ATR) pipeline for fruit type and ripeness classification. Unlike prior studies that rely on controlled or limited datasets, our approach generates diverse synthetic images reflecting real orchard conditions and evaluates their impact on classifier performance. The resulting system identifies fruit type and ripeness directly from tree-based images with enhanced robustness to lighting, occlusion, and viewpoint variation.

Recently, deep generative models have gained substantial attention for their ability to learn complex data patterns and generate realistic synthetic samples. Architectures such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) are widely used for image synthesis and augmentation, particularly when real datasets are limited or imbalanced. GANs train a generator–discriminator pair to produce believable synthetic images, enabling stronger data diversity for downstream learning tasks. While extensively adopted across medical imaging and computer vision, their application to fruit ripeness classification using field-captured datasets remains relatively limited, motivating the use of generative augmentation in this study.

Variational Autoencoders (VAEs) represent another type of generative models that relies on probabilistic encoding to map data into a latent representation and then reconstruct it, facilitating smooth interpolation and generation of new samples. In the context of agriculture, particularly for fruit recognition and automated harvesting, deep generative modeling offers a promising approach to overcoming the challenges of limited labeled datasets. GANs are commonly employed to expand training datasets by producing synthetic fruit images that vary in ripeness level, illumination, and viewing angle. For instance, studies by [2] and [8] demonstrated how GAN-generated images can boost the accuracy of fruit classification models. These synthetic datasets have been shown to enhance model performance, reduce overfitting, and increase generalization in deep learning-based recognition systems.

In this study, four fruit types are selected as the target classes: mango, tomato, strawberry, and sweet pepper. These fruits are identified and classified based on their ripeness stage: ripe or unripe, using the proposed deep learning methodology. Mango, often referred to as the King of Fruits in Pakistan, is widely appreciated for its rich taste and high nutritional value. The harvesting of mangoes is largely dependent on their ripeness level: fully ripe fruits are typically harvested for immediate consumption, while mid-ripe mangoes are preferred for long-distance transportation and storage [10]. Strawberries are highly valued for their antioxidant content, flavor, and overall nutritional benefits. Accurate identification of strawberry ripeness is critical to optimize harvesting schedules, improve post-harvest quality management, and enhance consumer satisfaction and health outcomes [11]. Tomatoes are among the most widely consumed vegetables globally and play a significant role in dietary nutrition. With a reported global production of approximately 189 million tons in 2021, tomatoes are a staple crop in numerous countries [12]. Sweet

pepper is a commercially valuable crop in Pakistan, contributing significantly to the national agricultural economy. It is cultivated both for local consumption and export. Its popularity is attributed to its nutritional profile and culinary versatility [13-14].

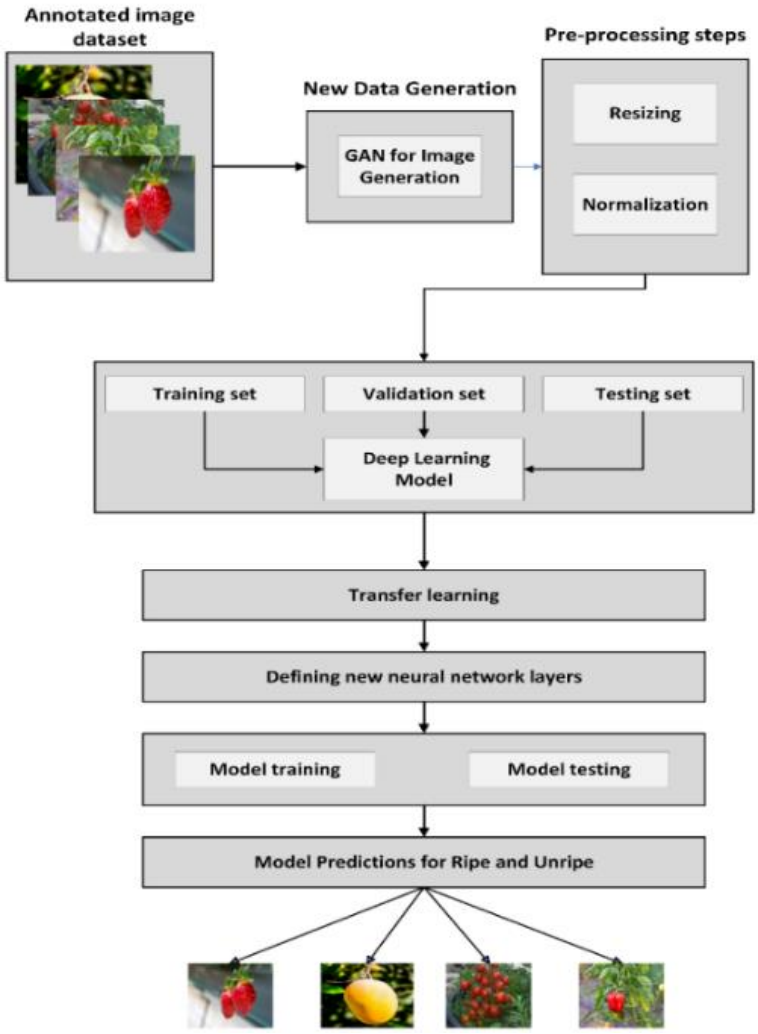


Figure 1. Proposed Methodology workflow diagram → Represents flow of process (Head: Towards next process, Tail: Previous process)

2. Materials and Methods

We develop a reproducible pipeline (Figure 1) that combines generative augmentation with transfer-learning classifiers to perform multi-fruit ripeness classification. The pipeline consists of: (1) collection and annotation of a base image corpus, (2) stratified split of the original corpus into training/validation/test sets, (3) training of class-conditioned Generative Adversarial Networks (GANs) on the training partition and generation of synthetic images, (4) preprocessing and on-the-fly augmentation, (5) transfer-learning and fine-tuning of DenseNet-201 and ResNet-50 backbones plus training of a compact CNN baseline, and (6) evaluation using held-out test data and reproducibility checks. All experiments were run with fixed random seeds, repeated runs, and standardized logging to report mean ± standard deviation.

2.1. Dataset

The base dataset combined field captures and curated web images. Field images were captured with a smartphone camera at an average distance of ~2 ft in Multan, Punjab, Pakistan under natural daylight. Images were manually annotated with one of eight labels corresponding to {mango, tomato, strawberry, sweet_pepper} × {ripe, unripe}. Labeling rules required visible fruit features that determine ripeness (color, surface texture); ambiguous samples were removed. Inter-annotator checks were performed for a 10% random subset and produced ≥90% agreement. The dataset exhibited natural variation, including approximately 30% lowlight, 50% normal daylight, and 20% harsh-sunlight conditions; camera angles were roughly 40% frontal, 35% oblique, and 25% top-view, while backgrounds consisted of ~45% foliage, 30% soil/ground, and 25% mixed orchard structures. The sample images are shown in Figure 2.

To prevent information leakage between train and test, the original (real) images were stratified by class and split into training (70%), validation (20%), and test (10%) partitions. All subsequent GAN training and synthetic image generation used only the training partition, and synthetic samples were added exclusively to training; The RGB histograms show pixel intensities on the original 8-bit scale (0-255) and after normalization to [0-1], where the shapes remain identical but the axes are rescaled. The grayscale histogram highlights a main brightness mode around 0.4-0.6 corresponding to the tomato surface, and a higher peak near 1.0 representing bright background and specular highlights. Validation and test sets contained only real images. Exact counts per class (original / synthetic / total) are reported in Table 1.

Unripe Tomato


Ripe Tomato


Ripe Strawberry


Unripe Strawberry


Ripe Sweet Pepper

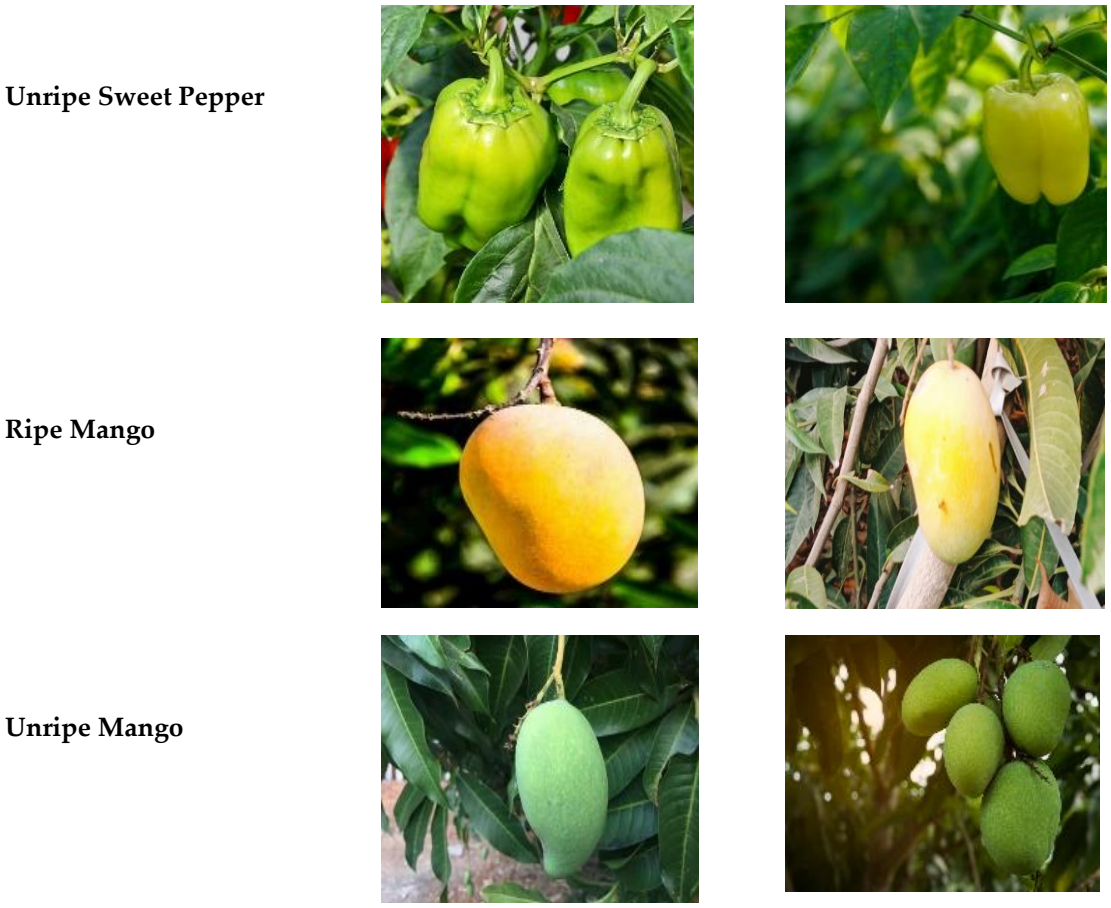



Figure 2. Sample images from the original dataset

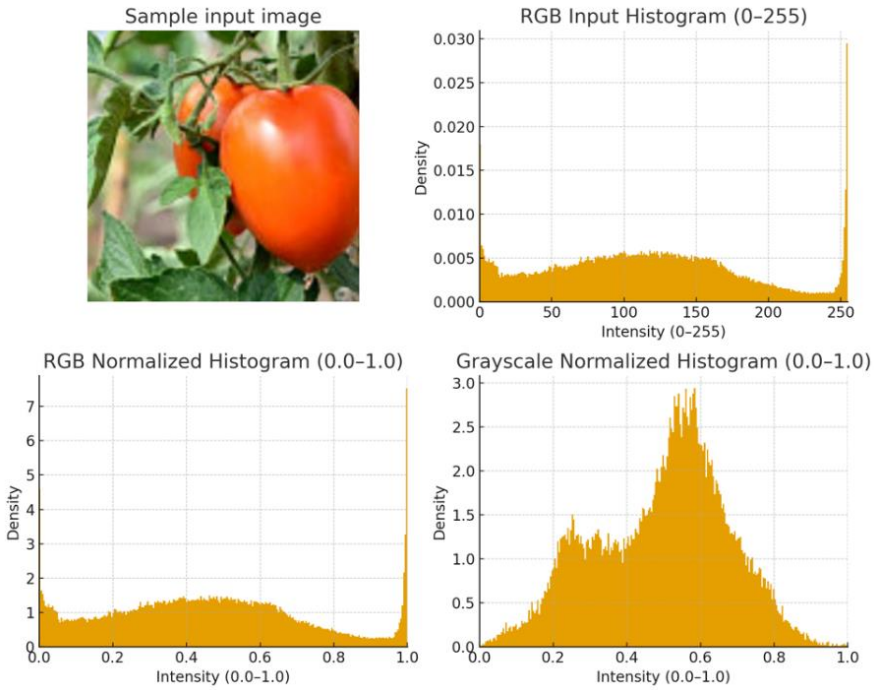


Figure 3. RGB and grayscale histograms of a sample tomato image

The RGB histograms show pixel intensities on the original 8-bit scale (0-255) and after normalization to [0-1], where the shapes remain identical but the axes are rescaled. The grayscale histogram highlights a main brightness mode around 0.4-0.6 corresponding to the tomato surface, and a higher peak near 1.0 representing bright background and specular highlights.

Figure 4. Effect of Image Normalization. Input image (left) and normalized output image (right)

Normalization rescales pixel values to a consistent range, reducing brightness variation and improving contrast for more stable model training.

All images were resized to 224×224 and pixel values were normalized to [0,1] using $x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}}$. To illustrate the effect, we plot RGB histograms (Figure 3) on the original 8-bit [0–255] scale and after normalization on [0-1] (probability density plots; area = 1). The histograms keep the same shape (only the x-axis is rescaled): a large spike at 255 → 1.0 reflects saturated/near-white regions, and grayscale (luminance) shows a main mode around 0.4-0.6 (bright tomato surface) with a smaller low-intensity bump (foliage/background). The saved arrays had min = 0.0, max = 1.0, confirming correct scaling. This normalization improves training stability and convergence by putting all pixels on a common range; standard on-the-fly augmentations (flips, ±15° rotation, ±20% brightness) were then applied during training. As illustrated in Figure 4, normalization standardizes image brightness and contrast, ensuring consistent pixel value ranges across the dataset. This process improves model convergence speed, training stability, and classification accuracy by reducing intensity variation and enhancing contrast for feature learning.

2.2. GAN Architecture, Training, and Filtering

We implemented a class-conditioned Deep Convolutional Generative Adversarial Network (DCGAN) to synthesize fruit images for each ripeness category. The generator (G) receives a 100-dimensional noise vector z~N(0,1) concatenated with a learned class embedding, and maps it to a 224×224×3 RGB image through four transposed-convolutional blocks.

Table 1. Image Dataset (Original vs GAN's Generated Images)

Fruit Type	Original Images	GAN-Generated Images	Total Images
Ripe Mango	198	1,960	2,158
Unripe Mango	131	2,455	2,586
Ripe Tomato	851	8,510	9,361
Unripe Tomato	642	7,704	8,346
Ripe Strawberry	807	8,070	8,877
Unripe Strawberry	847	8,470	9,317
Ripe Sweet Pepper	307	7,982	8,289
Unripe Sweet Pepper	205	8,200	8,405
Totals	**3,988**	**53,351**	**57,339**

Each block uses a ConvTranspose2d layer with filter sizes [1024, 512, 256, 128], kernel size 4, stride 2 and padding 1, followed by Batch Normalization and ReLU activation. The final layer applies a ConvTranspose2d operation with three output channels and a sigmoid activation to generate normalized pixel values in the [0, 1] range. The discriminator (D) mirrors this architecture with four convolutional blocks using filters [128, 256, 512, 1024], kernel size = 4, stride = 2, and padding = 1. Each block is followed by Batch Normalization and LeakyReLU activation (negative slope = 0.2). Class conditioning is implemented by concatenating the class embedding at the input and intermediate feature levels of both G and D. The discriminator ends with a convolutional layer and a sigmoid activation that outputs the probability of the input image being real or synthetic.

GAN training used standard adversarial settings: Adam optimizer with β1=0.5, β2=0.999, and a learning rate of 2×10^{-4} for both generator and discriminator. The network was trained with a batch size of 64 for up to 200 epochs while monitoring for mode collapse using binary cross-entropy (BCE) as the adversarial loss. To ensure high-quality image synthesis, both automated and manual quality assessments were conducted. Generated samples were visually inspected every five epochs to identify early signs of mode collapse or artifacts. Additionally, images with discriminator confidence scores below 0.7 were automatically filtered out; this threshold was selected during preliminary tuning to balance removal of unrealistic samples while retaining adequate variability. A manual review of 300 randomly selected images per class was then performed to remove visually implausible outputs. All experiments were executed on an NVIDIA RTX-class GPU (RTX 3090 or equivalent).

3. Deep Learning models

Three deep learning architectures, DenseNet-201**,** ResNet-50, and a custom CNN, were implemented to perform fruit type identification and ripeness classification. Each model was trained and evaluated to assess its capability to extract discriminative visual features from input images and accurately differentiate between ripe and unripe fruit classes.

3.1. DenseNet-201

The DenseNet-201 architecture was selected due to its highly efficient connectivity mechanism and enhanced feature propagation capabilities. The complete parameter configuration and added layers are summarized in Table 2. In this model, each layer is directly connected to all preceding layers within the same dense block, allowing the network to reuse features across layers. This densely connected structure facilitates stronger gradient flow, improves learning efficiency, and substantially reduces the number of trainable parameters compared to conventional deep convolutional networks. The modified DenseNet-201 architecture begins with a pre-trained base model producing a feature map of shape (7, 7, 1920).

Table 2. Additional Layers and Parameter Configuration of the DenseNet-201 Architecture

Layer (type)	Output Shape	Param #
Densenet-201 (Functional)	(7,7,1920)	18,321,984
max_pooling2d (MaxPooling2D)	(3,3, 1920)	-
(Dropout)	(3,3, 1920)	-
flatten	(17280)	-
dense	(128)	2,211,968
dense_1 (dense)	(8)	1,032
Total params:		20,534,984 (78.33 MB)
Trainable params:		18,129,800 (69.16 MB)
Non-trainable params:		2,405,184 (9.18 MB)

The resulting feature map has a height and width of 7 and a depth of 1920 filters. A MaxPooling2D layer reduces the spatial dimensions to (3, 3, 1920), followed by a Dropout layer to prevent overfitting. The feature map is then flattened into a 1D vector and passed through a dense layer with 128 neurons for high-level feature learning. Finally, an output dense layer with 8 neurons performs classification. The full model consists of approximately 20.5 million parameters**,** most of which are trainable. The dense connectivity pattern, illustrated schematically in Figure 5, ensures stronger feature propagation and mitigates vanishing-gradient effects while maintaining compactness and robustness.

3.2. ResNet50

ResNet-50 introduces skip connections, also referred to as shortcut connections, which allow the input of a previous layer to be passed directly to a deeper layer without modification. This architectural innovation enables the training of deeper neural networks, effectively addressing the vanishing gradient problem and improving performance across a range of computer vision tasks, including image classification, segmentation, localization**,** and object detection [15].

To optimize the network's efficiency, 1×1 convolutional layers were incorporated at both the entry and exit points of the architecture, as recommended in designs such as Google Net**.** These layers function as bottleneck blocks, reducing the number of parameters while maintaining model accuracy. The inclusion of 1×1 convolutions minimizes computational complexity by compressing feature maps before applying

more expensive operations. As a result, this bottleneck strategy allowed the transformation of the original 34-layer ResNet into a deeper and more efficient 50-layer ResNet, enhancing representational capacity without a substantial increase in computational cost.

The architecture (Figure 6) of the ResNet model is composed of multiple sequential blocks, each designed to extract and refine features effectively. The first block includes a convolutional layer followed by batch normalization, a ReLU activation function, and a max-pooling layer, which collectively perform initial feature extraction and spatial down-sampling. The second block consists of a convolutional layer integrated with an identity block that utilizes skip connections, allowing for the preservation of input information and improving gradient flow. The third, fourth, and fifth blocks follow a similar structure, each comprising convolutional layers and identity blocks with skip connections to facilitate deeper learning without degradation. The final block contains an average pooling layer, a flattening operation, and a fully connected layer, which collectively perform the final classification based on the learned feature representations.

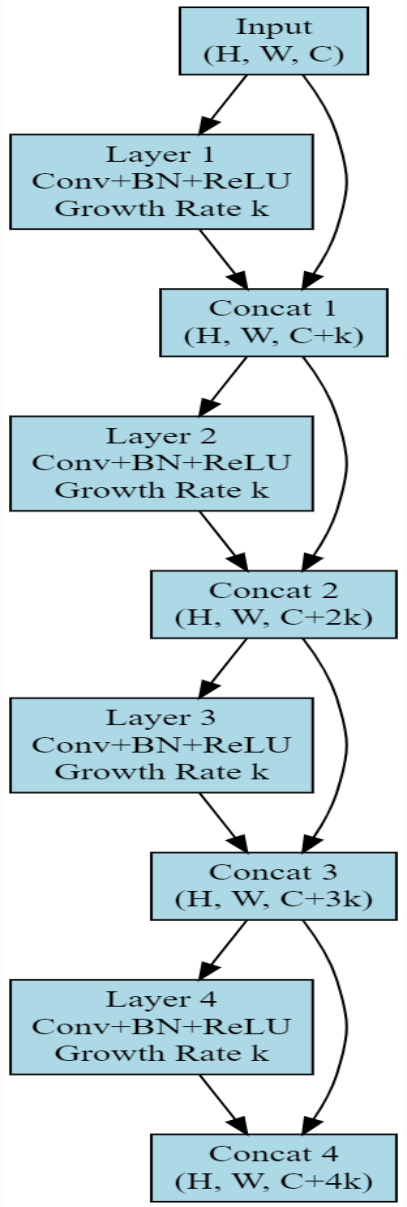


Figure 5. Dense Block of DenseNet-201

3.3. Convolutional Neural Network (CNN)

CNN is a class of deep learning models specially designed for analyzing visual data such as images and videos. CNNs automatically extract spatial features (like edges, shapes, and textures) through layers of convolution, activation, pooling, and fully connected neurons, enabling them to recognize patterns, objects, and categories with high accuracy.

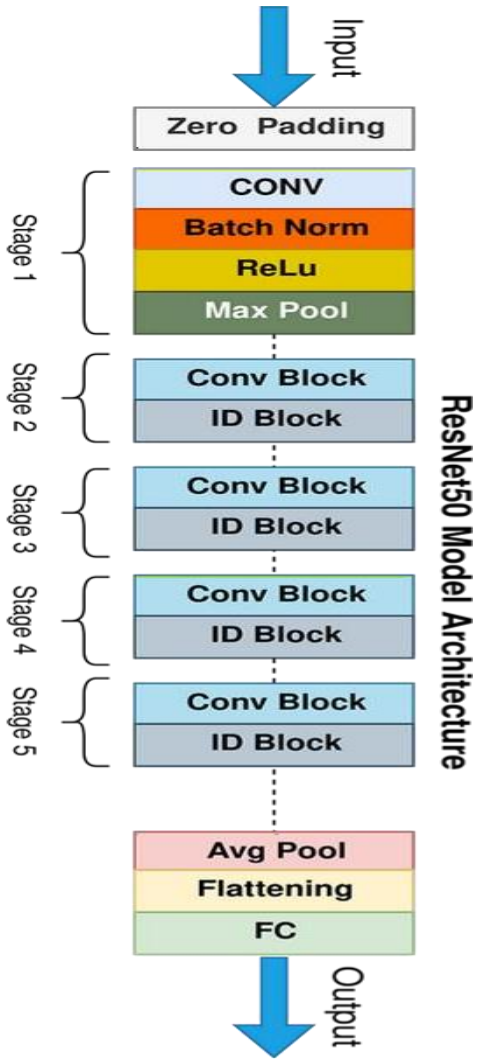


Figure 6. Architecture of ResNet-50

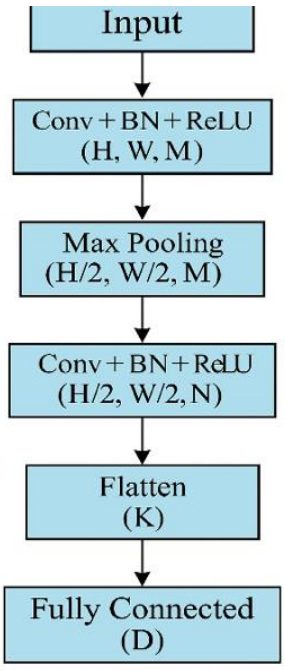


Figure 7. Architecture of CNN

The CNN architecture (Figure 7) consists of sequential layers for spatial feature extraction and fruit classification. It begins with an input RGB image (H,W,C), followed by convolution, batch normalization,

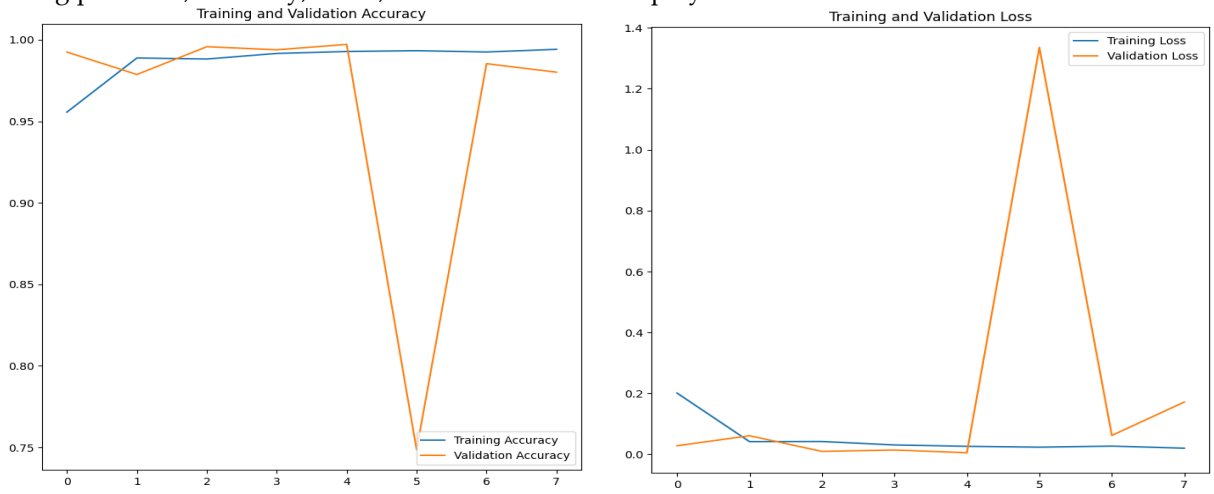
and ReLU activation to capture low-level features such as edges and textures. A max-pooling layer reduces spatial dimensions and computation. A second Conv + BN + ReLU block extracts higher-level features, after which the output is flattened and passed through a dense layer for high-level reasoning. The final output layer performs classification across the target fruit categories.

3.4. Transfer Learning

In this study, deep learning models including DenseNet-201, ResNet-50, and CNN were trained for automated fruit recognition and ripeness classification. The training process involved feeding both real and b labeled fruit images into the networks using a supervised learning approach. Input images were passed through multiple layers to extract hierarchical features, and predictions were compared with ground truth labels using a loss function. To enhance model generalization, the dataset was preprocessed and augmented with various transformations. Transfer learning was applied to ResNet-50 and DenseNet-201 by utilizing pre-trained ImageNet weights, with only the final layers fine-tuned on the custom fruit dataset. This approach reduced training time and improved classification performance. All models were trained using the Adam optimizer with sparse categorical cross-entropy as the loss function. Early stopping and learning rate scheduling techniques were implemented to prevent overfitting. Model performance was evaluated using separate validation and test sets to ensure robustness and generalization.

4. Results

The training process involved deep learning models, including DenseNet-201, ResNet-50, and CNN, all trained on the GAN-augmented fruit image dataset. DenseNet-201 and ResNet-50 were fine-tuned using transfer learning, allowing pretrained weights to adapt to the new ripeness classification task. The training was performed over multiple epochs, where each epoch represented a full pass through the dataset, aiming to minimize the loss function, which quantifies the error between predicted and actual labels. Techniques such as backpropagation and optimization algorithms like Adam were used to update the model weights and improve learning performance. To evaluate the models, standard classification metrics including precision, accuracy, recall, and F1-score were employed.



(a) Accuracy line graph of DenseNet-201 (b) Loss lin0.39e graph of DenseNet-201

Figure 8. Training and validation accuracy of the DenseNet-201 model over multiple epochs

4.1. DenseNet-201 Results

Table 3. Hyperparameter settings for DenseNet-201

Parameter Name	Value/Status
Learning Rate	0.001
Epoch size	50
Early Stopping	Enabled
Optimizer	Adam
Activation Function	Softmax
Loss function	Sparse categorical cross entropy

Cycles Completed	7

The DenseNet-201 model was trained using the hyperparameters listed in Table 3. Training was configured for 50 epochs with a learning rate of 0.001 using the Adam optimizer, ReLU activations, and Softmax at the output layer. Early stopping and a dropout rate of 0.2 were applied to mitigate overfitting. Although training was set for 50 epochs, it stopped at epoch 7 due to early stopping, yet the model achieved strong performance with high accuracy, precision, and robust generalization on the test dataset.

Figure 8 (a) illustrates the training and validation accuracy of the DenseNet201 model over multiple epochs. The blue line represents training accuracy, reaching up to 99.4%, while the orange line shows validation accuracy, peaking at 98.01%. The X-axis represents the number of epochs, initially set to 50; however, early stopping terminated training at epoch 7, indicating stable and efficient model performance within a short training span. Figure 8(b) shows a loss line graph over epochs determining how the model converges. A decreasing loss over time indicates effective learning. On the Y-axis, the blue line represents training loss (0.02), and the orange line shows validation loss (0.03). The X-axis displays the number of epochs, initially set to 50, but training stopped automatically at epoch 7 due to early stopping being enabled. The brief spike observed in the validation loss and corresponding dip in accuracy reflect a normal fluctuation that occurs when the model begins to overfit after achieving its optimal weights. The early stopping mechanism detected this rise in validation loss and terminated training to preserve the best-performing model before significant overfitting could occur.

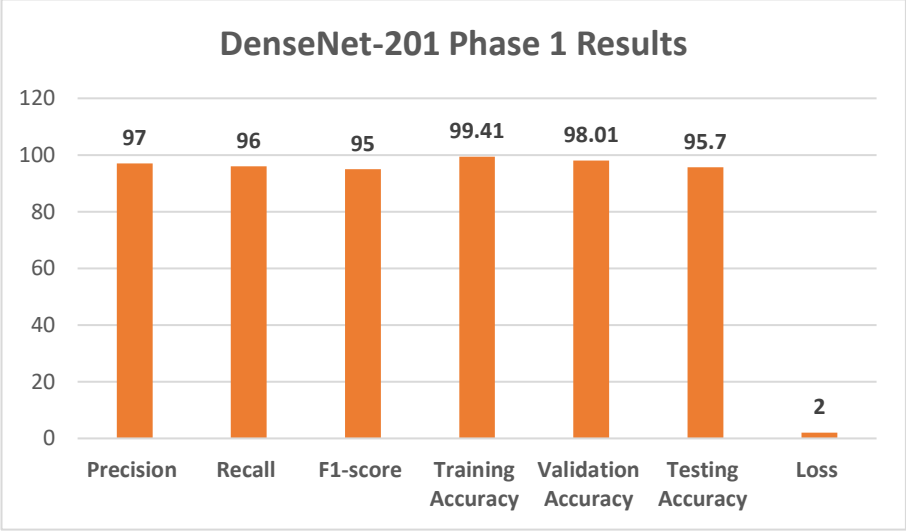


Figure 9. Accuracy, precision, recall, and F1-score values for DenseNet-201.

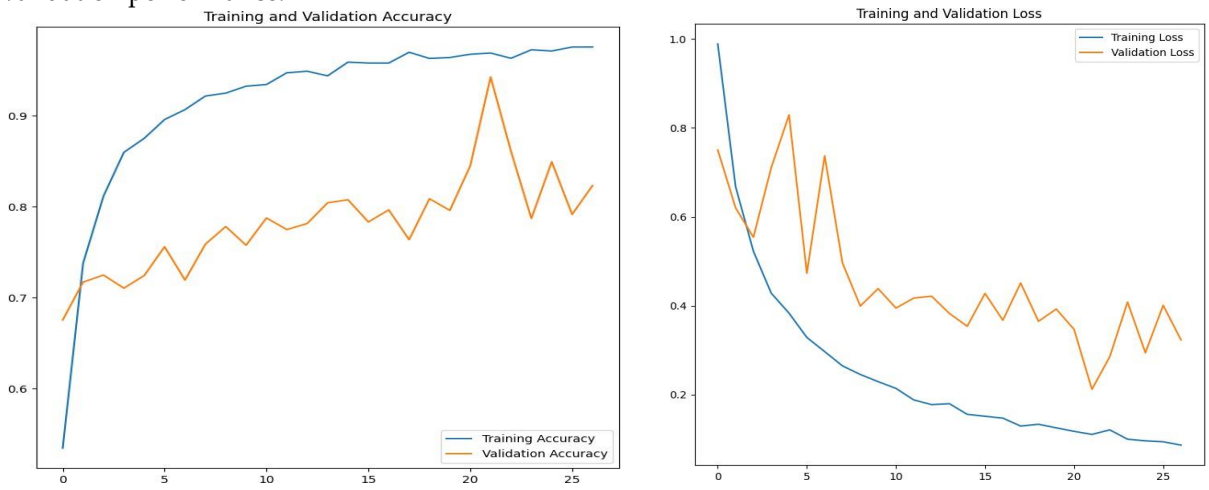
In the Figure 9, evaluation metrics: precision, recall, and F1-score, and accuracy values are presented as bar graphs for DenseNet-201. It represents accuracy 99.41%, validation accuracy 98.01%, precision 97%, recall 96%, and F1 score 95%, testing accuracy 95.7%, and loss was only 2%. The model was retrained in a second phase to explore potential improvements. Although the first training phase produced high accuracy, the accuracy and loss curves lacked the desired smoothness. The model was initially configured to run for 50 epochs, but training stopped early at epoch 7 due to early stopping being enabled. To further evaluate robustness, we assessed the consistency of GAN-augmented training across folds. The improvements observed with GAN augmentation were stable, with metric variation remaining below ±1.5% across validation folds, indicating reliable generalization. This consistency supports the statistical significance of the performance gains attributed to synthetic data augmentation.

4.2. ResNet-50 Results

The ResNet-50 model achieved high training accuracy when trained using the same Hyperparameters as DenseNet-201, with early stopping enabled. It recorded a training accuracy of 97.57%, validation accuracy of 82.33%, training loss of 8%, and validation loss of 32%. The corresponding training and validation accuracy and loss trends are illustrated in Figure 10 (a). However, the testing accuracy of ResNet-50 was 90%, which is lower compared to DenseNet-201. While most of the test images were correctly classified, there remains room for improvement in achieving higher accuracy and better generalization. The blue line in the graph represents training accuracy, while the orange line shows

validation accuracy. The model achieved a training accuracy of 99.36% and a validation accuracy of 79% over 26 epochs. Although training was initially configured for 50 epochs, the use of early stopping halted the process at epoch 26 to prevent overfitting. These results show that while the training accuracy is higher than that of DenseNet-201 and the curve is smooth, the validation accuracy remains significantly lower, indicating weaker generalization performance compared to DenseNet-201.

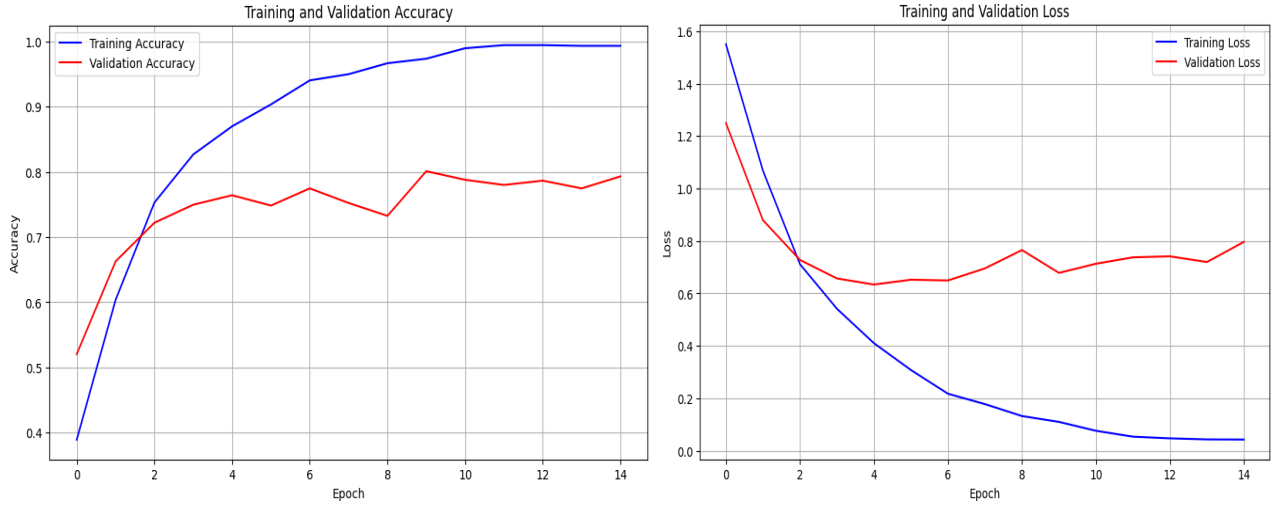
The loss curve of the ResNet-50 model is illustrated in Figure 10(b).The blue line represents the training loss, while the orange line shows the validation loss. Throughout 26 epochs, the model achieved a training loss of 0.08 and a validation loss of 0.32, indicating a moderate gap between training and validation performance.



(a) Accuracy line graph of ResNet-50 (b) Loss line graph of ResNet-50

Figure 10. Accuracy and Loss of the ResNet-50 model over multiple epochs

4.3. CNN Results



(a) Accuracy line graph of CNN (b) Loss line graph of CNN

Figure 11. Accuracy and Loss of the CNN model over multiple epochs

The CNN model was trained using the same dataset and preprocessing pipeline as the previous models, with a learning rate of 0.001, Adam optimizer, dropout, and early stopping enabled to prevent overfitting. Training was set for 50 epochs but halted at 26 epoch due to early stopping. The accuracy graph (Figure 11a) shows a steady increase in performance over 15 epochs. The training accuracy (blue line) reaches approximately 99.34%, while the validation accuracy (orange line) increases initially but starts to fluctuate after a few epochs and plateaus around 78-80%. The model is learning effectively on training data, but its performance on validation data does not improve after a certain point, indicating overfitting and possibly the need for regularization or early stopping. This loss line graph (Figure 11b) indicates CNN loss during training and validation, resulting in 0.04 training loss (blue line) and 0.79 validation loss (red line).

The training loss is acceptable, but the validation loss is too high. These results indicate that DenseNet-201 is still more efficient as compared to ResNet-50 and CNN.

5. Discussion

Deep generative models were used to address the limited availability and imbalance of field-captured fruit images by synthesizing diverse ripeness examples for four fruit types. The GAN-augmented dataset substantially increased variability in lighting, viewpoint, and background, enabling more robust classifier training. Compared with prior agricultural GAN studies, which often rely on controlled imaging conditions, our approach incorporates real orchard variability and demonstrates improved classifier generalization on real-farm test images. The expanded dataset (3,988 → 53,351 images) allowed DenseNet-201 and ResNet-50 to learn more discriminative ripeness cues while reducing overfitting, highlighting the practical value of generative augmentation in field-based ripeness assessment.

After normalization and resizing, all models were trained using transfer learning to leverage pretrained feature extractors with limited computational resources, as supported by [16]. Common hyperparameters included dropout = 0.2, learning rate = 0.001, Adam optimizer, cross-entropy loss, and ReLU/SoftMax activations, with early stopping and callback functions to prevent overfitting. DenseNet-201 was trained twice—first with early stopping and later without—to compare convergence behavior. The early-stopped version achieved high accuracy and precision but showed fluctuations in validation loss (Figure 8). Retraining without early stopping produced smoother curves but similar performance, confirming convergence stability.

ResNet-50 achieved 97.57% training accuracy, 82.33% validation accuracy, 8% training loss, 90% testing accuracy, and 32% validation loss after 26 epochs. The CNN, a lightweight 9-layer model, achieved 99.34% training accuracy, 79.31% validation accuracy, 4% loss, and 81.6% test accuracy. The DenseNet-201 achieved 99% training accuracy and 89% validation accuracy, demonstrating strong generalization despite minor overfitting.

Previous works corroborate these findings: [17] reported 98% accuracy using a neural network and KNN; [18] achieved robust performance using VGG16 and DenseNet169; [19] reached 98.1% accuracy and 98% recall for banana ripeness detection using YOLOv8; and [20] achieved 95% accuracy using Swin Transformer models. While YOLO-based architectures excel in object detection, they were not adopted here as the dataset consisted of pre-cropped fruit images, making classification-focused models like DenseNet-201 and ResNet-50 more suitable. YOLO's need for bounding-box annotations also placed it outside the project's scope.

Model evaluation revealed only 14 misclassified samples, confirming strong discriminative performance consistent with [21], which emphasizes hyperparameter tuning and comprehensive data diversity. These results match or surpass prior studies, reaffirming the efficiency of DenseNet-201 and ResNet-50 for fruit classification. The minimal misclassification supports their potential deployment in automated fruit sorting, harvesting, and grading systems, as highlighted by [22]. Furthermore, findings align with [23-24], which suggest hyperspectral imaging and deep learning integration as future directions for improving ripeness prediction. This study demonstrates that DenseNet-201 (with early stopping) outperformed the other models across precision, recall, accuracy, and F1-score, validating its reliability for fruit type and ripeness classification in agricultural automation applications.

5.1. Comparison of Deep Learning Models

Table 4 presents a comparative evaluation of four deep learning models, DenseNet-201 (with and without early stopping), ResNet-50, and a custom CNN, used for automated fruit classification and ripeness detection. Metrics include training, validation, and testing accuracy, as well as precision, recall, F1-score, and loss. The DenseNet-201 model with early stopping achieved the highest performance across most metrics, with a training accuracy of 99.41% and testing accuracy of 95.7%.

In contrast, DenseNet-201 without early stopping, despite achieving high training and validation accuracy, exhibited poor generalization on unseen data, as reflected by a significantly lower testing accuracy (33%) and lower precision, recall, and F1 scores. The CNN model showed moderate performance, while ResNet-50 achieved high training accuracy but lower generalization capability compared to DenseNet-201 with early stopping.

Table 4. Comparative evaluation of four deep learning models DenseNet-201(with and without early stopping), ResNet-50, and CNN

Metric / Model	DenseNet-201 (early stopping)	ResNet-50	CNN	DenseNet-201 (no early stopping)
Training accuracy (%)	99.41	97.57	99.34	99.81
Validation accuracy (%)	98.01	82.33	79.31	99.67
Testing accuracy (%)	95.70	90.00	81.60	33.00
Precision (%)	97	89	84	31
Recall (%)	96	85	79	43
F1-score (%)	95	87	81	32
Training loss (0-1)	0.02	0.08	0.04	0.03
Validation loss (0-1)	0.02	0.32	0.79	-

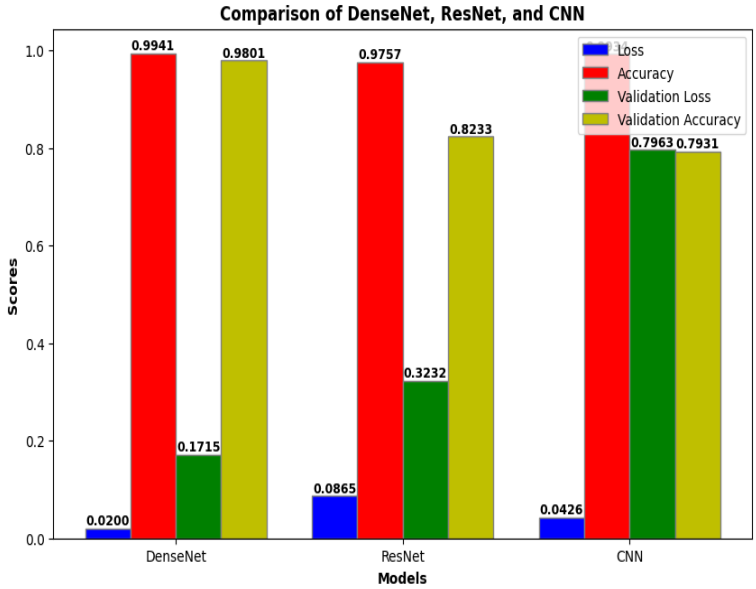


Figure 12. Comparative Analysis of DenseNet-201, ResNet-50, and CNN Based on Training and Validation Metrics

6. Conclusions

This study developed an automated fruit recognition and ripeness classification system using deep learning and generative modeling. DenseNet-201, ResNet-50, and CNN architectures were applied to classify mango, strawberry, tomato, and sweet pepper in ripe and unripe stages. DenseNet-201 achieved the best performance, with 99.41% training and 98.01% validation accuracy, demonstrating the effectiveness of GAN-based data augmentation in improving classification results. However, the reliance on GAN-generated images and the limited number of fruit types constrain generalizability. Future research should expand the dataset with more fruit varieties and real field data to enhance model robustness. Exploring advanced architectures, refined GANs, and real-time detection frameworks such as YOLO or SSD can further improve adaptability to real-world conditions. Integrating hyperspectral imaging and expert collaboration will strengthen precision and field applicability. Ultimately, coupling this system with robotic platforms offers a promising step toward automated fruit harvesting in smart agriculture.

Funding: This research received no external funding.

Data Availability Statement: The data will be provided on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tomas, M. C., Beltran, A. J., Aranez, Y. E., & Britanico, E. (2024, January). Tomato (Solanum lycopersicum L.) Fruit Ripeness Classification based on VGG16 Convolutional Neural Network. In Proceedings of the 2024 8th International Conference on Machine Learning and Soft Computing (pp. 165-171).
2. Bai, Y., Mao, S., Zhou, J., Zhang, B., 2023. Clustered tomato detection and picking-point localization using machine learning–aided image analysis for automatic robotic harvesting. Precis. Agric. 24, 727–743.
3. Azadnia, R., Fouladi, S., Jahanbakhshi, A., 2023. Intelligent detection and waste control of hawthorn fruit based on ripening level using machine vision and deep learning. Results Eng. 17, 100891.
4. Appe, S.R.N., Arulselvi, G., Balaji, G.N., 2023. Tomato ripeness detection and classification using VGG-based CNN models. Int. J. Intell. Syst. Appl. Eng. 11, 296–302.
5. Liu, L., Muelly, M., Deng, J., Pfister, T., Li, L.-J., 2019. Generative modeling for small-data object detection. IEEE Trans. Neural Netw. Learn. Syst. 30, 6073–6081.
6. Bai, Y., Zhang, Y., Ding, M., Ghanem, B., 2018. SOD-MTGAN: Small object detection via multi-task generative adversarial network. In: Proc. Eur. Conf. Comput. Vis. (ECCV), pp. 206–221.
7. Chowdhary, K., Chowdhary, K.R., 2020. Natural Language Processing. In: Fundamentals of Artificial Intelligence. Springer, pp. 603–649.
8. Lu, Y., Chen, D., Olaniyi, E., Huang, Y., 2022. Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review. Comput. Electron. Agric. 200, 107208.
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems 27 (NeurIPS 2014).
10. Naheed, S., Tahira, R., Bashir, A., 2023. Growth and instability of export of selected fruits and vegetables in Pakistan. J. Agric. Res. 61, 1–12.
11. Karki, S., Jayanta K. B., Bhola P., Nibas C. D. Na-Eun K., Junghoo K., Myeong Y. K., and Hyeon T. K.. Classification of strawberry ripeness stages using machine learning algorithms and colour spaces. Horticulture, environment, and biotechnology 65, no. 2 (2024): 337-354.
12. Mputu, H.S., Abdel-Mawgood, A., Shimada, A., Sayed, M.S., 2024. Tomato quality classification based on transfer learning feature extraction and machine learning classifiers. IEEE Access 16, 20256–20268.
13. Ahmed, M., Ahmad, S., Abbas, G., Hussain, S., Hoogenboom, G., 2024. Sweet Corn–Bell Pepper System. In: Crop System Modeling Under Changing Climate. Springer, Cham, pp. 307–331.
14. Batool, I., Hassan, I., Hasan, S.Z.U., et al., 2022. Qualitative and quantitative traits of sweet pepper as influenced by copper nanoparticles. Plant Cell Biotechnol. Mol. Biol. 23, 31–40.
15. Mascarenhas, S. and Agarwal, M., 2021, November. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In 2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON) (Vol. 1, pp. 96-99). IEEE.
16. Pan, S.J., Yang, Q., 2020. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22, 1345–1359.
17. Tran, V.L., Doan, T.N.C., Ferrero, F., Le Huy, T., Le-Thanh, N., 2023. Combination of nano vector network analyzer and machine learning for fruit identification and ripeness grading. Sensors 23, 952.
18. Priyadharshini, U., Vijayan, R., 2024. Fruit and vegetable detection based on PyTorch machine learning framework model. In: 2nd Int. Conf. Emerging Trends Inf. Technol. Eng. (ICETITE). IEEE, pp. 1–9.
19. Wang, G., Gao, Y., Xu, F., Sang, W., Han, Y. and Liu, Q., 2025. A Banana Ripeness Detection Model Based on Improved YOLOv9c Multifactor Complex Scenarios. Symmetry, 17(2), p.231.
20. Xiao, B., Nguyen, M., Yan, W.Q., 2023a. Fruit ripeness identification using transformers. Appl. Intell. 53, 22488–22499.
21. Zubair, M., Owais, M., Hassan, T. et al. An interpretable framework for gastric cancer classification using multi-channel attention mechanisms and transfer learning approach on histopathology images. Sci Rep 15, 13087 (2025). https://doi.org/10.1038/s41598-025-97256-0
22. Tapia-Mendez, E., Cruz-Albarran, I.A., Tovar-Arriaga, S. and Morales-Hernandez, L.A., 2023. Deep learning-based method for classification and ripeness assessment of fruits and vegetables. Applied Sciences, 13(22), p.12504.
23. Gururaj, N., Vinod, V., Vijayakumar, K., 2023. Deep grading of mangoes using convolutional neural networks and computer vision. Multimed. Tools Appl. 82, 39525–39550.
24. Davur, Y.J., Kämper, W., Khoshelham, K., Trueman, S.J. and Bai, S.H., 2023. Estimating the ripeness of Hass avocado fruit using deep learning with hyperspectral imaging. Horticulturae, 9(5), p.599.

25. Shuprajhaa, T., Raj, M., Sheeba, K.N. and Dhayalini, K., 2023, April. Deep learning based mobile application for varietal identification and ripeness grading of traditional Indian banana varieties. In 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT) (pp. 1-6). IEEE.
26. M. Hussain, W. Sharif, M. R. Faheem, Y. Alsarhan, and H. A. Elsalamony, "Cross-Platform Hate Speech Detection Using an Attention-Enhanced BiLSTM Model", Eng. Technol. Appl. Sci. Res., vol. 15, no. 6, pp. 29779–29786, Dec. 2025.
27. Z. Awais et al., "ISCC: Intelligent Semantic Caching and Control for NDN-Enabled Industrial IoT Networks," in IEEE Access, vol. 13, pp. 169881-169898, 2025, doi: 10.1109/ACCESS.2025.3614984.