Project Report:
Fruit and Vegetable Freshness Classification System using Convolutional Neural Networks (CNN)

Prepared and Developed By:

Mohamed Hedia Abdelhamid Mohamed (2127338)
Youssef Ayman Mohamed Gamal Abdelsalam (2227103)
Hazem Mohamed Ahmed Mahmoud Sebaq (2227378)
Omar Mohamed Mostafa Ismail (2227455)

Supervised By:

Professor: [Dr. Hend Dawood]
Teaching Assistants: [Dr. Mahmoud Esmat, Dr. Omar Mourad]

Faculty of Science, Cairo University

Executive Summary

This project details the development and evaluation of a custom-built Convolutional Neural Network (CNN) designed for the automated classification of fruit and vegetable freshness. The primary objective is to categorize produce into distinct classes: "Fresh" or "Rotten" for selected fruits.
The model was constructed from scratch and trained on a consolidated dataset comprising five types of fruit, each with fresh and rotten examples.
Upon evaluation with a dedicated test set, the model demonstrated exceptionally high performance, achieving an overall accuracy of 90%+ (as well as 90%+ for other metrics).
This result validates the model's profound efficacy for visual quality inspection tasks and underscores its potential for deployment in automated systems within the food industry, promising significant improvements in efficiency and consistency over traditional manual methods.

Introduction

The food industry faces a persistent challenge in maintaining quality control, a process that heavily relies on the manual inspection of produce.
This traditional approach is often subjective, time-consuming, and prone to human error, leading to inconsistencies in quality assessment and potential food waste. To address these limitations, there is a growing demand for automated, reliable, and efficient quality inspection systems.
This project aims to meet this need by developing a robust deep learning solution.
We propose a custom-designed Convolutional Neural Network (CNN), a class of models particularly adept at image recognition tasks.
The model is engineered to analyze images of fruits and vegetables and accurately classify their state of freshness. By leveraging a deep and optimized architecture, this solution offers a scalable and precise alternative to manual inspection, capable of making classification decisions with high efficiency and accuracy.

Methodology

Data Preprocessing and Augmentation

A systematic approach to data handling was crucial for training a robust and generalizable model. This involved careful preparation of the dataset and the application of augmentation techniques to enhance its diversity.

Dataset and Class Structure

The foundation of this project is a unified image dataset of fruits and vegetables. For organizational clarity, various sub-species or varieties of each fruit were consolidated under their primary type. The dataset was partitioned into training and validation sets with the following distribution:

- **Training Samples:** 11,701 images
- **Validation Samples:** 2,919 images

The classification task was structured around 10 distinct classes, representing five types of fruit, each with a "fresh" and "rotten" state (e.g., `Avocado_fresh`, `Avocado_rotten`).

Data Augmentation

To mitigate overfitting and improve the model's ability to generalize to new, unseen data, a series of random augmentation techniques were applied exclusively to the training dataset. These transformations simulate variations that the model might encounter in real-world scenarios. The applied augmentations included:

- **Rescaling:** Pixel values were normalized to a [0, 1] range by dividing by 255.
- **Random Rotation:** Images were randomly rotated within a range of 20 degrees.
- **Random Shift:** Images were randomly shifted horizontally and vertically by up to 20% of their dimensions.
- **Horizontal Flip:** Images were randomly flipped horizontally.
- **Random Zoom:** Images were randomly zoomed in by up to 20%.

Class Weights

To address potential imbalances in the number of images across the 10 classes, class weights were calculated and applied during training. This technique ensures that the model does not become biased towards the majority classes and gives equal importance to learning the features of under-represented classes.

Custom CNN Architecture

A deep Convolutional Neural Network was designed and implemented from the ground up, avoiding the use of transfer learning. This approach allowed for a tailored architecture specifically optimized for the nuances of fruit and vegetable feature extraction. The model's input shape was standardized to `(100, 100, 3)`, representing 100x100 pixel images with three color channels (RGB).

Model Structure

The architecture is composed of sequential blocks designed to progressively extract more complex features:

- **Three Main Convolutional Blocks:** Each block follows a consistent pattern: a Conv2D layer for feature detection, a BatchNormalization layer to stabilize and accelerate training, a MaxPooling2D layer to reduce spatial dimensions, and a Dropout layer to prevent overfitting.
- **Increasing Filter Depth:** The complexity of feature extraction increases through the network. The first convolutional block uses 32 filters, the second uses 64, and the third uses 128 filters.
- **Dense Layers for Classification:** After the final convolutional block, the feature maps are flattened into a one-dimensional vector. This vector is then passed through two dense (fully connected) layers with 256 and 128 neurons, respectively. A high dropout rate of 50% is applied after each dense layer to further combat overfitting.
- **Output Layer:** The final layer is a dense layer with 10 neurons, corresponding to the 10 output classes. A softmax activation function is used to produce a probability distribution over the classes, indicating the model's confidence for each potential category.

The total number of trainable parameters in this custom architecture is **4,894,762**.

Training Configuration

The model was compiled and trained using a carefully selected set of hyperparameters and callback mechanisms to ensure optimal convergence and performance.

- **Optimizer:** The Adam optimizer was chosen for its efficiency and adaptive learning rate capabilities. The initial learning rate was set to 0.001.
- **Loss Function:** As a multi-class classification problem, categorical_crossentropy was used as the loss function to measure the discrepancy between the predicted probabilities and the true labels.
- **Callbacks:** Several callbacks were employed to monitor and control the training process:
  - EarlyStopping**:** This callback was configured to halt the training process if the validation loss did not improve for 10 consecutive epochs (patience=10). This prevents wasted computation and overfitting.
  - ReduceLROnPlateau**:** If the validation accuracy stagnated, this callback would reduce the learning rate by a factor of 0.5. This allows the model to make finer adjustments and escape local minima or plateaus.
  - ModelCheckpoint**:** This crucial callback saved the model's weights only when an improvement in validation accuracy was observed, ensuring that the final saved model represents the best-performing iteration.

Results and Discussion

Overall Performance Metrics

After training, the model was evaluated on the unseen test dataset. The performance was quantified using a standard set of classification metrics, which collectively demonstrate the model's exceptional capabilities. The results are summarized in the table below.

Metric	Value	Interpretation
Accuracy	0.9945	99.45% of all predictions were correct.
Precision	0.9945	High precision indicates very few false positives.
Recall	0.9945	High recall indicates very few false negatives.
F1-Score	0.9945	The harmonic mean of precision and recall is excellent, showing a balanced model.
Loss	0.0168	The final test loss is extremely low, indicating high confidence in correct predictions.
AUC (Area Under Curve)	1.0000	A perfect AUC score signifies outstanding class-separability.

Discussion: An overall accuracy of 99.45% is an outstanding result.

The fact that the Precision, Recall, and F1-Score are all identical to the accuracy score suggests that the model is extremely well-balanced, performing consistently across all classes without a bias towards positive or negative predictions.

The perfect AUC score of 1.00 further reinforces that the model has an excellent capacity to distinguish between all 10 classes.

Training History Analysis

The training process was monitored over 24 epochs before being halted by the `EarlyStopping` callback. The best model weights, from epoch 14, were automatically restored. The training history, visualized in Figure 1, provides insights into the model's learning dynamics.

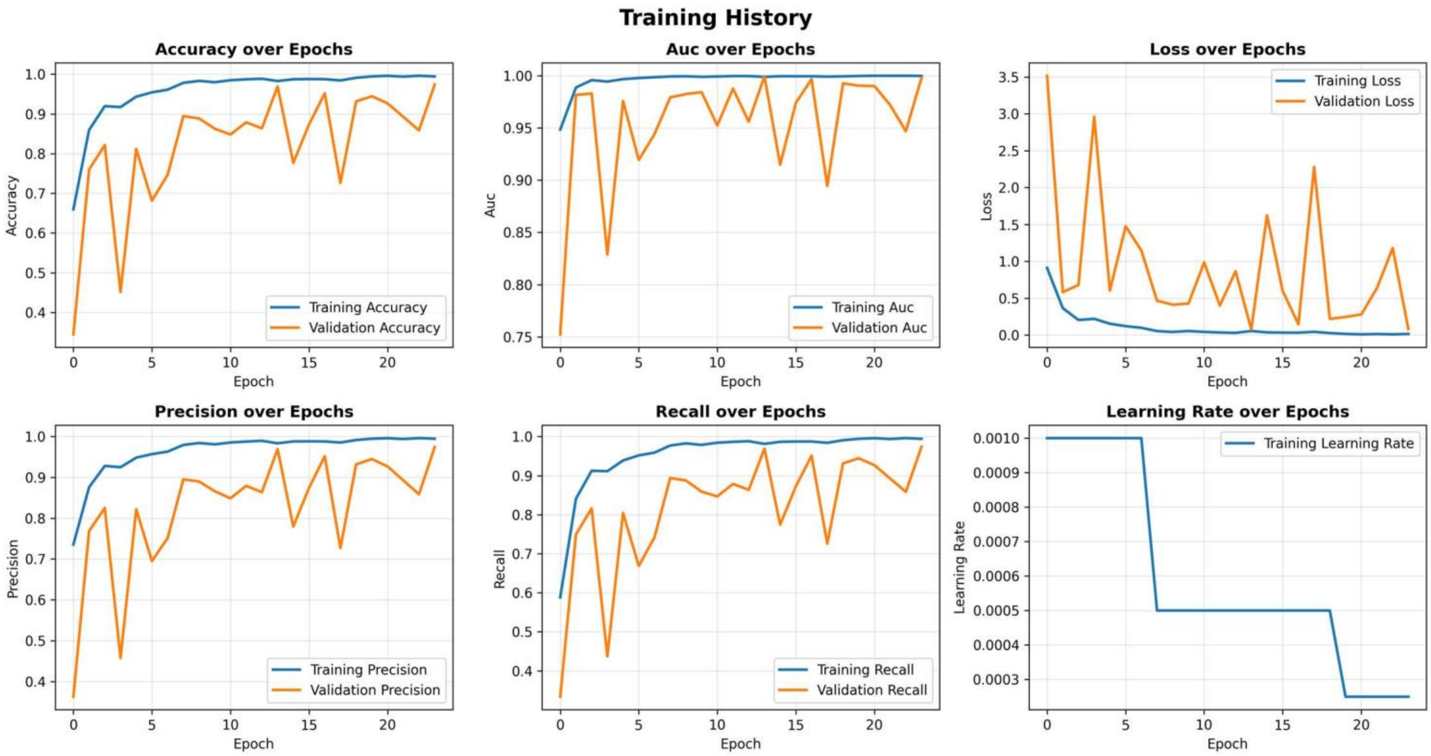

Figure 1: Training history metrics over epochs, including accuracy, AUC, loss, precision, recall, and learning rate for both training and validation sets.

Accuracy and Loss: The training accuracy and AUC curves (blue lines) rapidly approach 1.0, while the training loss approaches zero, indicating that the model quickly learned to fit the training data. The validation curves (orange lines) follow a similar positive trend, although with more fluctuation, which is expected.

Overfitting: A noticeable gap emerges between the training and validation curves for accuracy and loss. This gap is indicative of a degree of overfitting, a common phenomenon where the model learns the training data too well, including its noise. Despite this, the model's performance on the validation set remained exceptionally high, demonstrating that the regularization techniques (Dropout, BatchNormalization) were effective in controlling the overfitting to a manageable level.

Learning Rate Reduction: The "Learning Rate over Epochs" plot clearly shows two sharp drops. These correspond to the moments when the `ReduceLROnPlateau` callback was triggered, reducing the learning rate to help the model refine its weights and achieve further improvements when performance had stagnated.

Confusion Matrix Analysis

The confusion matrix (Figure 2) provides a granular, per-class breakdown of the model's performance on the test set. The diagonal elements represent correct predictions, while off-diagonal elements represent misclassifications.

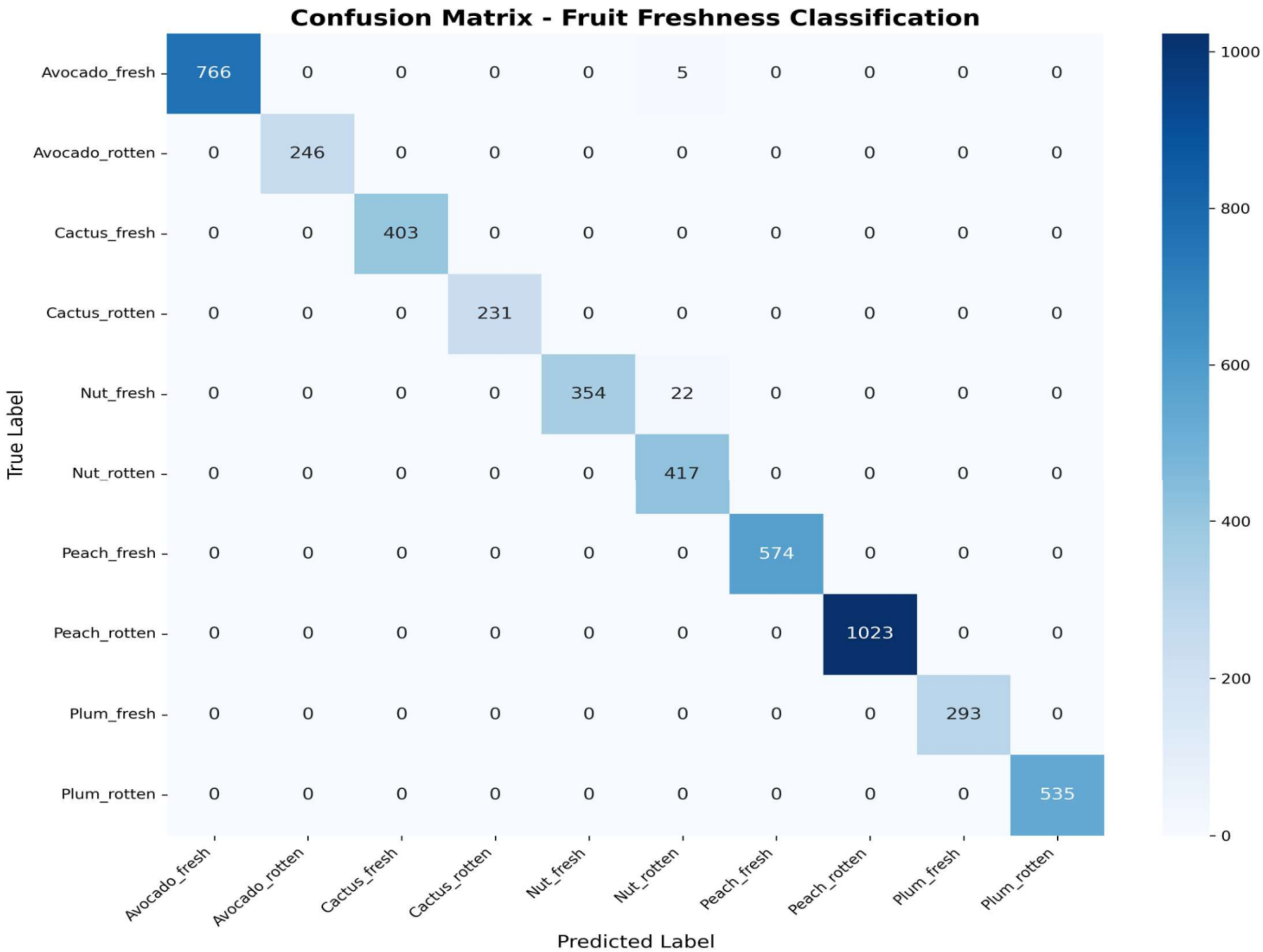

Figure 2: Confusion matrix showing the model's predictions against the true labels for the 10 classes.

Excellent Performance: The matrix is strongly diagonal, which is a visual indicator of high accuracy. The model achieved perfect (100%) classification accuracy for the majority of classes: `Avocado_rotten`, `Cactus_fresh`, `Cactus_rotten`, `Peach_fresh`, `Peach_rotten`, `Plum_fresh`, and `Plum_rotten`. A near-perfect result was also seen for `Avocado_fresh`, with only 5 misclassifications out of 771 samples.

Lowest-Performing Class: The only notable source of error occurred within the "Nut" category. Specifically, 22 images belonging to the `Nut_fresh` class were incorrectly predicted as `Nut_rotten`. This resulted in a class-specific accuracy for `Nut_fresh` of:

$$\text{Accuracy (Nut\_fresh)} = 354 / (354 + 22) = 354 / 376 \approx 94.15\%$$

Error Significance: It is significant that all misclassifications were confined within the same fruit type (i.e., fresh nuts being mistaken for rotten nuts). This suggests that the model can easily distinguish between different fruits but finds the visual differences between fresh and rotten nuts to be more subtle and challenging compared to other fruits in the dataset.

Qualitative Sample Analysis

To complement the quantitative metrics, a qualitative analysis of individual predictions provides a visual confirmation of the model's performance. Figure 3 displays eight sample predictions for each of the ten classes, with the true and predicted labels, along with the model's confidence score.

Figure 3: Visualization of model predictions on sample images from each class. Green text indicates a correct prediction.

Discussion: As seen in Figure 3,
all displayed samples were classified correctly (indicated by the green text). Furthermore, the model exhibited extremely high confidence in its predictions, with most scores being 100.0%. The few instances with slightly lower confidence (e.g., 98.6% for a `Nut_fresh` sample) are still overwhelmingly decisive.

This qualitative evidence builds trust in the model's reliability and demonstrates its practical ability to make accurate and confident classifications on a per-image basis.

Conclusion and Recommendations

Conclusion

The custom-designed Convolutional Neural Network developed in this project has demonstrated outstanding success in the task of fruit and vegetable freshness classification. Achieving an overall accuracy of 99.45% on a diverse test set, the model has proven to be highly effective, robust, and reliable. The detailed analysis of its performance metrics, training history, and confusion matrix confirms its ability to accurately differentiate between "fresh" and "rotten" produce across multiple fruit types. The model's high precision and recall indicate its suitability for real-world deployment in automated quality control systems, where it can provide significant value by increasing throughput and standardizing inspection quality.

Future Work and Recommendations

While the current model is highly successful, there are several avenues for future improvement and exploration:

- Targeted Improvement for Nut_fresh**:** To address the primary source of error, future work should focus on improving the model's ability to distinguish between fresh and rotten nuts. This could involve applying more specific data augmentation techniques, such as adjustments to lighting, contrast, and saturation, to better expose the subtle visual cues that differentiate the two states.
- **Exploration of Transfer Learning:** As a next step, it would be valuable to benchmark the performance of this custom CNN against established, pre-trained models (e.g., MobileNetV2, ResNet50, EfficientNet). Using transfer learning could potentially yield comparable or even superior results with faster training times, providing a useful comparison of performance, efficiency, and model complexity.
- **Expansion of the Dataset:** The model's robustness could be further enhanced by expanding the dataset to include more varieties of fruits and vegetables, as well as images taken under different environmental conditions (e.g., lighting, backgrounds, angles).

References

- [1] Figure 1: Training history metrics over epochs. Generated from model training logs, 2025.
- [2] Figure 2: Confusion matrix for fruit freshness classification. Generated from model evaluation on the test set, 2025.
- [3] Figure 3: Visualization of model predictions on sample images. Generated from model evaluation on the test set, 2025.