**Lightweight Convolutional Neural Networks for CAVENDISH BANANA DECAY ASSESSMENT: A Knowledge Distillation Approach for Edge Deployment**

**STUDENT NAME:** Otinwa Ayomide

**STUDENT NUMBER:** C00313536

**COURSE NAME:** Masters in Applied Artificial Intelligence

**DEPARTMENT:** Department of Computing

**COURSE CODE:** ZDISC5201

**WORD COUNT:** 10899

**SUPERVISOR:** Rejwanul Haque

**DATE OF SUBMISSION:** 20 08 2025

**PLAIGIAISM DECLARATION**

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or processional qualification except as specified.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SIGNATURE DATE

**TABLE OF CONTENTS**

[**1. INTRODUCTION** 4](#_Toc206425756)

[**2. LITERATURE REVIEW** 5](#_Toc206425757)

[**3. MODEL ARCHITECTURE** 7](#_Toc206425758)

[3.1 Teacher Model 7](#_Toc206425759)

[3.2 Student Model 8](#_Toc206425760)

[3.3 Knowledge Distillation Framework 9](#_Toc206425761)

[3.4 Activation Function 10](#_Toc206425762)

[3.5 Evaluation Metrics 11](#_Toc206425763)

[**4 METHODOLOGY** 12](#_Toc206425764)

[4.1 Dataset Collection and Preprocessing 12](#_Toc206425765)

[4.2 Teacher Model Training 12](#_Toc206425766)

[4.2 Student Model Training 13](#_Toc206425767)

[**5. RESULTS** 14](#_Toc206425768)

[**6. MODEL CONVERSION TO TENSORFLOW LITE** 16](#_Toc206425769)

[**7. DEPLOYMENT** 17](#_Toc206425770)

[7.1 Web Application Interface 17](#_Toc206425771)

[7.2 AWS Integration and Configuration 18](#_Toc206425772)

[**8. ETHICAL AND SOCIETAL CONSIDERATIONS** 18](#_Toc206425773)

[**9. ERROR ANALYSIS** 19](#_Toc206425774)

[**10. LIMITATIONS OF THE STUDY** 20](#_Toc206425775)

[**CONCLUSION** 20](#_Toc206425776)

[**FUTURE WORK** 20](#_Toc206425777)

[**ACKNOWLEDGMENTS** 21](#_Toc206425778)

[**REFERENCES** 21](#_Toc206425779)

[**Appendix 1: DATA DEFINITIONS** 24](#_Toc206425780)

[**Appendix 2: TURNITIN REPORT** 25](#_Toc206425781)

**Lightweight Convolutional Neural Networks for Cavendish Banana Decay Assessment: A Knowledge Distillation Approach for Edge Deployment**

Ayomide Otinwa  
 Applied Artificial Intelligence  
 South East Technological University, Carlow  
Carlow, Ireland  
 c00313536@setu.ie

***Abstract*-** Bananas are among the most consumed fruits worldwide but are highly perishable, and have a shelf life of only a few days, which contributes significantly to food waste. Ripeness assessment relies on human discerning, which is inconsistent and prone to a lot of errors. This dissertation presents a lightweight deep learning framework for automated banana ripeness classification using knowledge distillation. A large ResNet152 teacher model (200 MB) was used to transfer both hard-label accuracy and soft-label probability distributions to a significantly smaller ResNet10 student model. The model attained an accuracy of 97.12%, closely approximating the teacher model’s 97.75%, while being significantly smaller. After converting the model to TensorFlow Lite, its size was reduced to 7.5 MB, showing its suitability for deployment on low-resource devices.

The model was integrated into a web application designed for small retailers and vendors with limited computational capacity. Users can upload banana images, receive immediate ripeness predictions, and store both results and images in the cloud for future retraining. This work demonstrates the potential of knowledge distillation for resource-aware agricultural AI solutions.

Keyword­­­- Banana ripeness classification, knowledge distillation, deep learning, TensorFlow Lite, web application, Teacher Model, Student Model

# **1. INTRODUCTION**

Food wastage remains a critical global challenge, with perishable agricultural produce such as bananas contributing significantly to the problem. Bananas, especially, have a short shelf life of approximately 5 to 7 days (DETSI, 2023), in which their ripening process accelerates rapidly. And if they are not consumed or processed appropriately, this accelerated ripening often leads to premature spoilage. The result of this wastage goes beyond economic losses, but also impacts food security (Speare-Cole, 2025) and increases environmental concerns due to the additional resources consumed in their production, transportation, and disposal. Accurate assessment of banana ripeness stages can directly address this issue by enabling better decision-making in supply chain management, storage strategies, and planning for consumption.

Usually, ripeness evaluation is performed manually, which relies on human judgment. While this approach can be fast, it is subjective and susceptible to variability based on the assessor’s experience, lighting conditions, and fatigue. These inconsistencies are a problem for businesses that depend on accurate grading, particularly small-scale retailers, local market vendors, street fruit sellers, and small grocery store owners, who often sell bananas individually rather than in large commercial batches. For these sellers, even a small percentage of misjudged fruit can result in significant profit losses, reduced customer trust, and unnecessary waste.

In recent years, computer vision with the aid of deep learning has acted as a good solution for automating the classification of fruit maturity, offering the potential to replace personalized visual inspection with accurate, reproducible, and data-driven assessment. Convolutional Neural Networks (CNNs) have shown very good capability in various applications for agriculture, including fruit quality detection, disease diagnosis, and ripeness classification (Tapia-Mendez et al., 2023). However, these high-performing models typically demand substantial computational resources in terms of processing power and memory, making them impractical for deployment on devices with limited hardware capabilities such as edge devices like the Raspberry Pi.

To overcome these limitations, this research utilizes knowledge distillation, which is a model compression technique where a large, complex, and high-performing “teacher” model transfers its learned knowledge to a smaller, lightweight “student” model (Bergmann, 2024). The process enables the student model to retain much of the teacher’s accuracy while drastically reducing size, memory requirements, and inference time. This approach is particularly well-suited for low-computation devices, where quick responses and low energy consumption are essential. By reducing the computational requirements, knowledge distillation makes it possible to deploy accurate ripeness detection tools in resource-constrained environments, thus serving users who lack access to such technology.

In this work, a high-capacity teacher network is first trained to accurately classify bananas into four ripeness categories, after which its knowledge is distilled into a small student network. This student model is integrated into a custom-built, single-page web application tailored for non-technical users. The application provides a clean and intuitive interface where sellers can upload an image of a banana, either captured through a phone camera or uploaded from the device storage, and instantly receive a classification result along with a confidence score. To ensure that the solution not only aids real-time decision-making but also contributes to continuous improvement, the system is integrated with Amazon Web Services (AWS) S3. Every uploaded image, along with its predicted label, is stored in a centralized cloud bucket, creating a growing archive of labeled data that can be used for model retraining, seasonal adaptation, and performance monitoring.

The main contribution of this study lies in showing how knowledge distillation can be applied to fruit ripeness detection, producing a lightweight yet accurate model suitable for environments with limited computational resources. By compressing a large teacher model into a student, this work answers the research question of whether model compression techniques can maintain strong performance while enabling deployment on edge devices. Also, the project investigates the challenges of visually overlapping ripeness stages, offering insights into error patterns and model behavior that can guide future research.

Overall, this project represents a practical application of deep learning to an everyday problem faced by small-scale retail sellers. By combining deep learning with model compression, cloud-based storage, and an intuitive interface, it offers a scalable and low-cost means of improving the accuracy of banana ripeness assessment. In doing so, it has the potential to reduce post-harvest losses, optimize sales strategies, and encourage more sustainable fruit distribution practices, which benefits not only individual sellers but also the broader agricultural supply chain.

# **2. LITERATURE REVIEW**

The detection of fruit ripeness and decay using computer vision has evolved rapidly over the past decade, driven by the aid of deep learning advancements, increased availability of agricultural datasets, and the demand for automation in food quality assessment. As global food supply chains face mounting pressure to reduce waste, improve grading accuracy, and enable efficient sorting, researchers have explored numerous neural network architectures and deployment strategies to balance accuracy, computational cost, and scalability. These are recent works in fruit classification, decay detection, and knowledge distillation, with a focus on the methods and findings directly relevant to the development of a compact, accurate banana ripeness classifier deployable in real-world settings.

One of the most consistent findings across the literature is the effectiveness of lightweight CNN architectures for agricultural classification tasks. (Ashik et al., 2025) developed a binary fruit freshness classification system using MobileNetV2 trained on an extensive dataset of over 40,000 images. The model achieved 95.66% accuracy while remaining computationally efficient, making it highly suitable for edge deployment scenarios. The authors highlighted the importance of real-time inference and low memory requirements, priorities that align closely with the objectives of this work. Similarly, (Suharjito et al., 2021) and (Chakraborty et al., 2021) reinforced the potential of MobileNetV2 in agricultural vision tasks, with both studies reporting competitive or superior performance compared to heavier architectures such as InceptionV3 and VGG16, while requiring only a fraction of the parameters. These findings emphasize the value of using compact architectures in situations where resources are limited, such as on mobile devices or single-board computers.

While classification remains the dominant approach, some researchers have expanded their focus to pixel-level analysis of decay through segmentation techniques. (Stasenko et al., 2021) applied U-Net, DeeplabV3, and Mask R-CNN to segment decayed zones in stored apples, achieving 98.81% mAP for whole apple detection and 43.6% mAP for decayed areas. Their work demonstrated that fine-grained spatial information can provide additional insights into decay severity and distribution, which could be beneficial for early detection. However, these methods often require high computational resources, making them less feasible for embedded deployment. This trade-off between precision and deployability reinforces the need for model compression strategies like knowledge distillation.

Knowledge distillation was first popularized by (Hinton et al., 2015), and has appeared as a main method for model compression without significant accuracy loss. (Xu et al., 2024) illustrated its agricultural application by creating a pest detection framework in which a YOLOv5l teacher model guided a YOLOv5s student. By introducing multi-scale attention fusion and a focal entropy loss function, they transferred both classification and localization knowledge, boosting the student’s mAP by 3.2% compared to baseline while keeping inference times suitable for real-time deployment. This is directly relevant to the present work, where a large ResNet152 teacher is used to train a much smaller ResNet10 student for efficient banana ripeness classification.

Similarly, (Wang et al., 2025) demonstrated that distillation can effectively transfer accuracy from a large EfficientNet-B7 teacher to a MobileNetV3 student in defect classification of green plums, reducing computational cost without degrading performance significantly. The distilled MobileNetV3 retained 94.85% accuracy compared to the teacher’s 95.67%, with notable gains in inference speed and suitability for edge devices. These results matched the goals of this project for preserving high classification performance while enabling practical, real-time deployment in the field.

Further enhancement of distilled models can be achieved by incorporating attention mechanisms into both teacher and student networks. In the context of apple leaf disease diagnosis, researchers integrated a coordinate attention module into a ResNet34 teacher and MobileNetV2 student, achieving 99.31% accuracy for the student which is only 0.22% lower than the teacher (Dong et al., 2024). This demonstrated that targeted attention not only improves feature extraction but also facilitates more effective knowledge transfer. Banana-specific research provides important domain insights into the classification of ripeness stages under varying real-world conditions. (Shu et al., 2025) leveraged YOLOv5 to classify bananas into four categories under different lighting and viewing angles. With a mean average precision of 98.6% and real-time inference speeds of 30 FPS, their work validated the feasibility of deploying such models in sorting lines. However, their reliance on a relatively large YOLOv5 architecture leaves room for improvement in low-resource contexts. (Baldovino et al., 2024) widened this concept using YOLOv8-S to classify bananas into six ripeness stages. While they achieved strong static image accuracy (89.5%), they encountered common misclassification challenges between ripe and overripe stages, as well as reduced multi-object detection performance in live webcam streams. These limitations highlight the value of strategies like dataset balancing and distillation, both of which are employed in this work to address stage confusion.

Outside of bananas, (Pariya et al., 2024) explored multi-fruit decay detection using a custom CNN that outperformed architectures such as VGG16, ResNet, and AlexNet. Their model proved how flexible it was under challenging conditions such as occlusion and extreme lighting, real-world scenarios where banana classification models must also perform reliably. Importantly, the robustness of their approach under data scarcity aligns with the philosophy of distillation.

The diversity and quality of training datasets are crucial for model generalization. (Sultana et al., 2022) introduced a diversified dataset containing fresh and spoiled fruits, as well as lighting and background variations, which improves the effectiveness of model training in less controlled environments. Similarly, studies by (Miah et al., 2021) and (Aksoy et al., 2024) on different architectures aimed at classifying fresh and spoiled products provided benchmarks for model selection before distillation. The general performance trends show that some lightweight networks may still gain from distillation, even after being optimized for running on mobile devices.

(Chen and Chu, 2020) applied self-distillation to fine-grained mango grading, incorporating triplet and ordinal losses to handle ordinal label relationships. This approach, which improved classification accuracy without requiring a separate teacher network, could be adapted where class labels also exhibit ordinal structure. Meanwhile, Transformer-based architectures such as FastSegFormer (Cai et al., 2024) have integrated distillation for lightweight semantic segmentation in agriculture. While these models achieve impressive accuracy-speed trade-offs, their relative complexity and dependency on specialized hardware currently limit their adoption in low-cost deployment scenarios.

Several research studies have shown that the optimization of edge devices is both feasible and essential to the efficient operation of agricultural AI systems. (Cai et al., 2024) proved that segmentation distilled networks can attain comparable performance to a teacher model with processing rates in excess of 70 frames per second, thus enabling real-time agricultural applications. Similarly, (Ashik et al., 2025) and (Suharjito et al., 2021) point out that the right choice of architecture, training approaches, and deployment platforms will yield efficient models appropriate for devices with limited processing and memory capabilities. The findings from these studies largely dictate the methodology of this current study, which involves training a high-capacity ResNet152 teacher model, which is then distilled to produce a lower-capacity ResNet10 student model, with further optimization carried out using TensorFlow Lite in preparation for deployment.

In summary, the literature reflects a clear track towards balancing accuracy, efficiency, and deployability in agricultural computer vision systems. Lightweight architectures such as MobileNet and ShuffleNet have repeatedly proven effective, while knowledge distillation has proved to be a powerful tool for retaining accuracy in compressed models. The methods and findings of these prior works collectively form the foundation for the approach taken in this research.

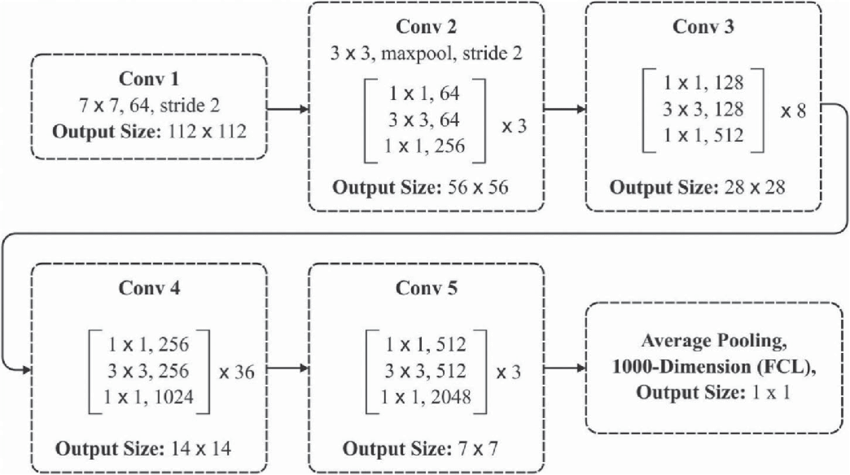
# **3. MODEL ARCHITECTURE**

The architecture for this research project is built on the principle of knowledge distillation. The approach used is response-based distillation, in which the student learns not only from the hard ground truth labels but also from the softened probability outputs of the teacher. This allows the student to capture inter-class relationships while maintaining a lightweight structure suitable for efficient inference.

## 3.1 Teacher Model

The teacher model in this study was built using the ResNet152, one of the deepest convolutional neural networks in the Residual Networks family. It has 152 layers and stands as a high-capacity architecture designed for large-scale image recognition tasks, and it is widely recognized for its performance on the ImageNet challenge (image-net, 2017). One of the distinctive characteristics of ResNet152 is that it utilizes deep residual connections on a large scale, allowing it to learn residual rather than indirect mappings. These skip connections effectively mitigate the vanishing gradient problem, enabling the stable training of extremely deep networks that would otherwise suffer from accuracy degradation as layers’ increase. This capability makes ResNet152 not only accurate but also highly reliable in extracting discriminative features from complex datasets.

The architecture of ResNet152 is composed of an initial 7×7 convolutional layer with stride 2, followed by batch normalization, ReLU activation, and max pooling to establish a coarse spatial representation. Beyond this, the network is structured into 50 bottleneck residual blocks, organized into four major stages. Each bottleneck block employs a combination of 1×1 and 3×3 convolutions: the 1×1 layers act as dimensionality reduction and expansion layers, lowering computational cost, while the 3×3 layers capture fine spatial details. Together, this bottleneck design achieves both depth and efficiency. Throughout the architecture, batch normalization layers enhance training stability and reduce internal covariate shift, while the final global average pooling layer compresses high-dimensional feature maps into compact representations for classification. These design choices make ResNet152 particularly effective at identifying subtle visual patterns, such as the varying textural and color transitions that occur across banana ripeness stages.



**Fig. 1. Architecture of the ResNet152 teacher model. The figure highlights the convolutional layers and bottleneck residual blocks that enable deep feature extraction.**

In this research, ResNet152 was not used as a classifier but rather as a feature extractor. The model was initialized with pre-trained ImageNet weights, which allowed it to leverage generalized visual features, and its original classification head was removed. The base convolutional layers were frozen throughout training to preserve the pre-learned hierarchical representations, while the extracted features were passed into a custom classification head tailored for the ripeness detection. This added head contained a global average pooling layer along with three fully connected dense layers with 256, 128, and 64 neurons. Each dense layer incorporated ReLU activation, batch normalization, and a 50% dropout rate, enhancing regularization and reducing the risk of overfitting to the relatively smaller domain-specific dataset. The final output layer contained four units, corresponding to the categories unripe, ripe, overripe, and rotten, with logits generated for multi-class classification.

The ResNet152 was chosen as the teacher model because of its ability to generate sophisticated, detailed discriminative features from images. These features made it well-suited for detecting subtle textural and color differences between ripeness stages. Within the framework of response-based knowledge distillation, these features are distilled into the lighter ResNet10 student network, enabling the student to approximate the teacher’s performance while operating at a fraction of the computational cost. In this way, the teacher model acts as a knowledge reservoir, providing both accurate class boundaries and varied inter-class relationships that are challenging for smaller models to learn independently.

## 3.2 Student Model

The student model used in this study was ResNet10 [8], a compact residual network specifically chosen for its balance between representational power and computational efficiency. While substantially smaller than the ResNet152 teacher model, which contains 58.9 million parameters, ResNet10 maintains the core philosophy of residual learning introduced in (He et al., 2015). This ensures that the model benefits from stable gradient flow and improved convergence, even with significantly fewer layers. At roughly 1.98 million parameters, ResNet10 is over 95% smaller compared to the ResNet152, making it an excellent candidate for deployment in edge devices.

The architecture starts out with a 7×7 convolutional layer having a stride of 2, followed by batch normalization and ReLU activation, which provides the foundation for feature extraction. A 3×3 max pooling layer further reduces the spatial resolution, making sure there is efficient down sampling before the network enters the deeper residual stages. These stages are structured into four groups of basic residual blocks, each composed of two 3×3 convolutional layers with batch normalization and ReLU activations. Each block includes identity skip connections that pass the input directly to the output, preserving information and alleviating the vanishing gradient problem. When a change in dimensionality occurs, the skip connection is adjusted using a 1×1 convolution with batch normalization to match the new dimensions.

Gradually, the network expands its feature representation capacity by increasing the number of filters while reducing spatial resolution, starting with 64 filters in the first stage, 128 in the second, 192 in the third, and 256 in the fourth. This gradual scaling allows the network to get both low-level texture details and higher-level semantic features relevant to the ripeness classification. Following the residual stages, a global average pooling layer aggregates the feature maps into a compact representation, which is then fed into a fully connected dense layer with four output units corresponding to the ripeness categories.

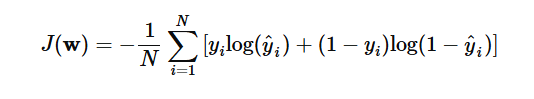
The ResNet10’s design makes it particularly effective for the purpose of this study. On one hand, its lightweight structure allows for fast inference and low energy consumption, which are important factors in resource-limited deployment scenarios. On the other hand, its residual learning framework ensures that despite its simplicity, the network is capable of modeling the subtle visual patterns associated with the different ripeness stages. Moreover, in the context of knowledge distillation, ResNet10 benefits from supervision not only through the hard ground truth labels but also with the soft probability distributions provided by the ResNet152 teacher. This form of guidance enables the student to learn both class-discriminative features and the subtler inter-class relationships embedded in the teacher’s predictions, ultimately improving their generalization performance without sacrificing efficiency.

## 3.3 Knowledge Distillation Framework

The framework in this study was guided using response-based knowledge distillation, where the student network learns from the ground-truth labels and the soft probability distributions generated by the teacher model. Normally, supervised training relies solely on hard labels, where each image is assigned a discrete class. While this approach is effective for clear-cut distinctions, it often fails to capture the relative similarities between classes, which is critical in fine-grained classification problems. In contrast, response-based distillation provides the student with the softened output logits of the teacher, which encode richer information about class relationships. This dual approach ensures that the student model receives both the precise, ground-truth information required for correct classification and the subtler insights about the inter-class relationships that the teacher has learned through extensive training. This knowledge is particularly valuable in cases where visual differences between categories are subtle. To achieve this, the training combines two complementary loss functions.

The first is Sparse Categorical Cross-Entropy (SCCE), which directly measures the difference between the student’s predicted class, which is expressed as integer indices and the true class labels (GeeksforGeeks, 2025). SCCE is particularly efficient in scenarios with mutually exclusive classes, as it focuses on minimizing the prediction error for the exact ground-truth label. The Sparse Categorical Cross-Entropy loss is expressed as:

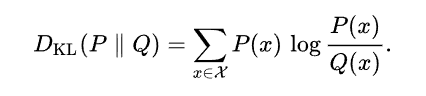
(1)



Here, w represents the model parameters (including the neural network weights), yᵢ denotes the true integer label for the ith sample, and (ŷᵢ ) + (1 - yᵢ )log(1 - ŷᵢ ) is the probability assigned by the student model to the correct class. This loss enforces that the student’s predictions converge as closely as possible to the ground truth provided in the training set.

The second loss function is Kullback–Leibler Divergence (KL Divergence), which transfers the teacher’s soft targets, which are its probability distributions over all classes when predictions are made using a temperature parameter greater than one. KL Divergence measures the difference between the softened probability distributions of the teacher and student, allowing the student to capture not only which class the teacher predicts but also how confident it is relative to other classes. The KL Divergence loss for soft targets is defined as:

(2)



Where P and Q are the teacher and student's softened probability distributions over X classes. This process allows the student model to learn from the correct output class, and also the relative similarities between classes obtained by the teacher model. This dual-loss framework is implemented through a custom Distiller class.

The Distiller class, which is implemented as a custom Keras function, manages this dual-loss training process, updating only the student’s parameters while keeping the teacher’s parameters fixed. By combining the hard label learning with soft target learning, the student model benefits from the supervision and the teacher’s embedded knowledge, leading to a more improved generalization and robustness, particularly for edge deployment. Each training step calculates the SCCE loss between the student’s predictions and the ground truth, alongside the KL Divergence loss between the student’s and teacher’s soft predictions. These losses are then combined into a weighted sum, with a hyperparameter α controlling the trade-off between hard-label accuracy and soft-label knowledge transfer. A temperature parameter T is also used during the softening process, ensuring that probability distributions reveal finer distinctions between classes.

By integrating both hard label learning and soft target learning, the training process allows the student model to leverage the structured knowledge embedded in the teacher’s decision boundaries. This results in a model that not only matches the teacher’s classification performance more closely but also exhibits improved generalization and robustness in deployment. For resource-constrained edge environments such as small retail stores, this approach provides a powerful balance between accuracy, speed, and hardware efficiency, making real-time banana ripeness detection both practical and accessible.

## 3.4 Activation Function

The Rectified Linear Unit (ReLU) has become the most widely used activation function in deep learning, particularly in convolutional neural networks. It is defined mathematically as:

f(x)=max(0,x) (3)

This simple formula outputs the input directly if it is positive and zero otherwise, introducing non-linearity while maintaining computational efficiency. Unlike activation functions such as the sigmoid or hyperbolic tangent, which saturate for large positive or negative inputs and thereby suffer from the vanishing gradient problem (Srivastava, 2024), ReLU avoids this issue for positive values. This allows models to converge faster during training and enables the training of much deeper networks without degradation in performance.

A notable advantage of ReLU is its ability to promote sparse activations since negative inputs are mapped to zero, and only a subset of neurons activate at a time. This sparsity improves both computational efficiency and model generalization (Nair and Hinton, 2010). However, ReLU has its limitations such as the dying ReLU problem, which occurs when neurons consistently output zero due to negative inputs and effectively stop updating during training. To address this, several types such as Leaky ReLU, Parametric ReLU (PReLU), and Exponential Linear Units (ELU) have been used, which allow small non-zero outputs for negative inputs to maintain gradient flow (He et al., 2015).

Despite these challenges, ReLU remains the standard choice in many applications because of its balance between simplicity, performance, and scalability. Its use was important in enabling breakthroughs in large-scale computer vision tasks, such as the ImageNet classification challenge, where networks using ReLU achieved really high accuracy while being significantly faster to train than those using other activation functions.

In summary, ReLU provides the necessary non-linearity for neural networks to learn complex functions, facilitates efficient gradient propagation, and ensures computational practicality, making it a foundation for modern deep learning architectures.

## 3.5 Evaluation Metrics

To assess the performance of classification models, this study uses widely recognized evaluation metrics such as the accuracy, precision, recall, F1-score, and the confusion matrix. These metrics are all derived from four fundamental outcomes in classification problems:

True Positives (TP): Cases where the model accurately identifies an instance as part of a specific class.

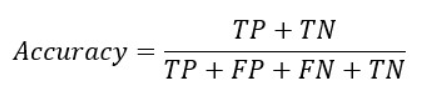
True Negatives (TN): Cases where the model correctly recognizes that an instance does not belong to a particular class.

False Positives (FP): Cases where the model mistakenly labels an instance as belonging to a class when it actually does not.

False Negatives (FN): Cases where the model fails to identify an instance as part of a class when it actually belongs.

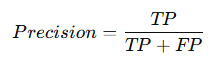
1. **Accuracy**: represents the ratio of all correct predictions (both TP and TN) to the total number of predictions:

(4)



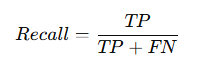
1. **Precision**: measures the proportion of correctly predicted positive instances out of all instances predicted as positive.

(5)



1. **Recall**: measures the model’s ability to capture all relevant positive instances. It is defined as the ratio of true positives to all actual positives.

(6)



1. **F1-Score**: combines precision and recall into a single metric by calculating their harmonic mean.

(7)



The harmonic mean is used instead of the arithmetic mean because it penalizes extreme differences between precision and recall.

1. **Confusion Matrix**: provides a structured way to visualize the performance of a classification model. For binary classification, it is a 2×2 matrix containing TP, TN, FP, and FN values. For multi-class problems, the confusion matrix extends to an n×n matrix, where n is the number of classes.

Each row of the matrix represents the actual class. Each column represents the predicted class. The diagonal elements show correctly classified samples, while the off-diagonal elements show misclassifications.

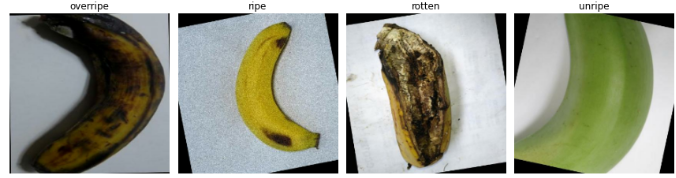
# **4 METHODOLOGY**

## 4.1 Dataset Collection and Preprocessing

The dataset used for this study was sourced from Roboflow Universe (Roboflow Universe Projects, 2025), with a public repository carefully curated for banana ripeness classification. The dataset was already augmented and labeled, and the images have been classified under four different ripeness stages: unripe, ripe, overripe, and rotten. These augmentations included horizontal and vertical flips to simulate different viewing perspectives, as well as rotations of 90° (clockwise, counter-clockwise, and upside down) alongside free rotations within a range of –15° to +15° to account for natural variations in fruit orientation. Zooming and cropping were applied at up to 20% to mimic differences in image scale, while adjustments in hue, saturation, brightness, and exposure within ±10% were incorporated to replicate lighting inconsistencies commonly encountered in real-world environments. Additionally, a slight blur of up to one pixel was introduced to simulate minor image quality degradation, and all images were resized to 416×416 pixels during export to maintain consistency for model input.

The total number of images contained in the dataset was 11,793, with a non-uniform count across the four classes. To ensure class balance throughout the training process, a custom Python script was developed to upsample every class to a specific target size. This was achieved by randomly duplicating images within each class to meet the predefined thresholds:

1. Training Set: 5,000 images per class (20,000 total)
2. Validation Set: 400 images per class (1,600 total)
3. Test Set: 200 images per class (800 total)



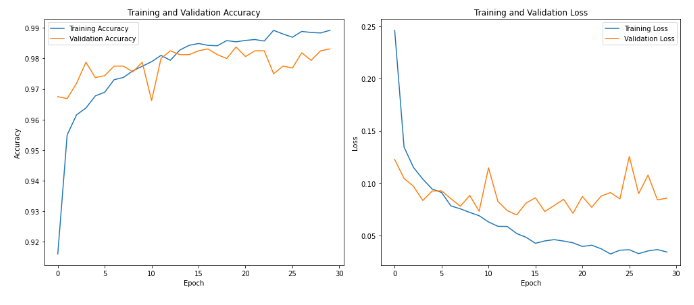
**Fig. 1. Augmented Images of Banana across the four ripeness stages**

All images were resized to 224×224 pixels using TensorFlow's image loading pipeline to match the input requirements of the CNN architecture. The images were normalized to the [0, 1] range by dividing the pixel values by 255.0. The dataset was loaded using TensorFlow’s image\_dataset\_from\_directory() utility, which allows for automatic labeling based on the structure of the folder. The dataset was organized into three directories (train/, valid/, and test/), each containing subfolders named after the corresponding class labels. A batch size of 32 was used for the training and evaluation. Shuffling was applied to the training dataset to minimize overfitting and ensure randomness during training. This preprocessing ensured that the model was trained on a balanced, diverse, and realistic dataset, capable of adapting to varying conditions such as lighting, orientation, and background noise, which are commonly encountered in real-world settings.

## 4.2 Teacher Model Training

The ResNet152 teacher model was initialized with weights that were pre-trained on the ImageNet dataset. The top classification layer was removed to allow the model to adapt to the task of ripeness classification. During training, the base of the model was frozen in order to preserve the generalizable feature representations learned from ImageNet while also reducing computational demands. To customize the model for the four ripeness stages, 3 additional fully connected layers were added on top of the frozen backbone. These consist of a 256-unit dense layer followed by batch normalization and a dropout layer with a 50% rate, then a 128-unit dense layer also paired with batch normalization and dropout, and a 64-unit dense layer preceding the final classification layer. The last layer contained four output neurons, each corresponding to one of the target classes.

The optimizer used was the Adam Optimizer, and a batch size of 32. The Sparse Categorical Cross entropy loss function effectively handled the integer-encoded class labels, and the model performance was evaluated with sparse categorical accuracy, reflecting how well the model predicted the correct class. To prevent overfitting, early stopping was used with a patience of five epochs by monitoring validation loss. This ensured that training would halt once the model stopped improving on unseen data, with the best weights restored. Training continued for up to 30 epochs on the dataset, while the validation set helped to track the model’s ability to generalize.



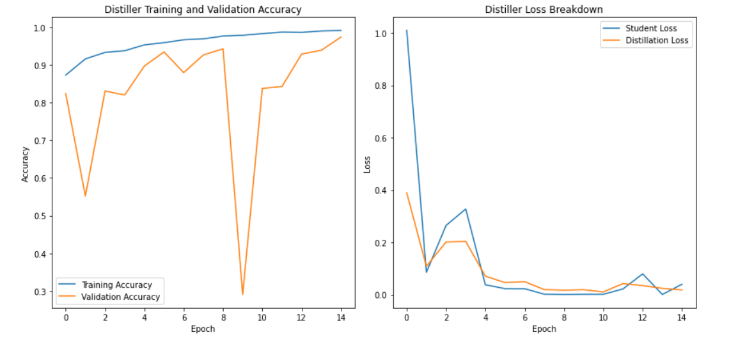
**Fig. 2. Training accuracy and loss curves of the ResNet152 teacher model. The figure shows how accuracy improves and loss decreases over epochs, demonstrating stable convergence during training.**

The training history, shown in Figure 2, shows a smooth and consistent improvement in both training and validation accuracy. The model started with a training accuracy of 91.61% and validation accuracy of 96.75% in the first epoch, steadily improving to 98.92% training accuracy and 98.31% validation accuracy by the final epoch. The loss curves show a corresponding decline, with training loss decreasing from 0.2461 to 0.0344 and validation loss reducing to 0.0858, indicating strong convergence without severe overfitting.

There were Minor fluctuations in validation accuracy and loss toward the later epochs, but overall, the curves demonstrate close alignment between training and validation performance, implying good generalization to unseen data. The final evaluation on the test set confirmed this, with the model achieving 97.75% accuracy. This established a strong performance baseline, making it an effective candidate for guiding the training of the smaller student model.

## 4.2 Student Model Training

The student model uses the ResNet10 architecture, which is a significantly smaller variant of the ResNet family designed to reduce computational complexity while retaining the residual learning mechanism that improves gradient flow during training. The network was constructed with an initial convolutional layer using a 7 \* 7 kernel and a stride of 2, followed by batch normalization, ReLU activation, and max pooling. This was followed by four residual stages, each composed of a basic residual block with two 3 \* 3 convolutions and batch normalization layers. Down sampling was performed at the beginning of stages two to four using a 1 \* 1 convolution in the skip connection to match dimensions. The final stage output was passed through a global average pooling layer before being fed to a fully connected layer with four output neurons corresponding to the banana categories. The student model was trained over 15 epochs using the distillation framework described earlier, with α=0.05 and a temperature T=2. The Adam optimizer and a batch size of 32 were employed for efficient gradient-based updates. Throughout training, the sparse categorical accuracy on the training set exhibited a steady improvement, rising from 87.29% in the first epoch to 99.12% by the final epoch. This indicated that the model was able to rapidly understand both the hard labels from the dataset and the softened target distributions from the teacher.



**Fig. 3. Training accuracy and loss curves of the ResNet10 student model. The figure illustrates how the lightweight network progressively improves in performance while maintaining stability, highlighting the effectiveness of knowledge distillation in guiding its learning**

The validation accuracy followed a more variable trajectory, starting at 82.38%, dipping at certain points (notably epoch 2 with 55.19%) due to batch-specific variability, before recovering strongly and achieving 97.37% in the final epoch. The corresponding loss curves show that the student loss decreased consistently, while the distillation loss followed a similar downward trend, confirming that the network increasingly aligned with the teacher’s soft predictions over time. Also, certain validation epochs showed sharp fluctuations in accuracy and loss, such as the large drop in accuracy at epoch 10 (29.06%) which corresponded to a sudden spike in validation loss (26.78). These anomalies are attributable to the sensitivity of small validation subsets to class distribution. However, the general convergence pattern and final performance metrics suggest that the distillation setup effectively transferred representational knowledge from the ResNet152 teacher to the compact ResNet10 student.

The training and validation performance are visualized in Figure 3, which presents both accuracy and loss curves. The accuracy graph shows a general upward trend, with the validation curve ultimately matching the training curve in the later epochs. The loss graph reveals a smooth decline for the training loss and a more variable pattern for the distillation loss, though both stabilized toward the end of training. This alignment between training and validation metrics reinforces the model’s ability to generalize to unseen data despite its reduced complexity.

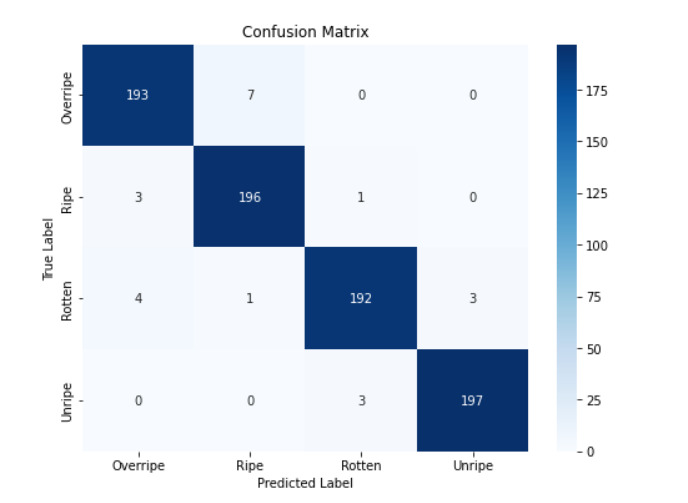
# **5. RESULTS**

The ResNet152 teacher model achieved a classification accuracy of 97.75% on the test dataset, which shows its effectiveness in learning the discriminative features for the four ripeness. The classification report shows consistently high precision, recall, and F1-scores across all classes, with values exceeding 0.96 in every metric. This shows a balanced performance and minimal bias toward any particular class.

Table. 1. Classification Report for ResNet152 model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-score | Support |
| Overripe | 0.96 | 0.96 | 0.96 | 200 |
| Ripe | 0.96 | 0.98 | 0.97 | 200 |
| Rotten | 0.98 | 0.96 | 0.97 | 200 |
| Unripe | 0.98 | 0.98 | 0.98 | 200 |
| Accuracy |  |  | 0.97 | 800 |
| Macro avg | 0.97 | 0.97 | 0.97 | 800 |
| Weighted avg | 0.97 | 0.97 | 0.97 | 800 |

The confusion matrix Figure 4 further illustrates the model’s performance, with the majority of predictions falling along the diagonal, indicating correct classification. Misclassifications were minimal, with the largest being seven overripe bananas misclassified as ripe. Other errors were scattered and infrequent, suggesting that the model distinguishes well between the four stages despite visual similarities in borderline cases.



**Fig. 4. Confusion matrix of the ResNet152 teacher model. The figure shows the distribution of correct and incorrect predictions across the four ripeness categories, reflecting the model’s high discriminative power and minimal misclassification.**

Overall, these results confirm that the ResNet152 teacher model provides a strong baseline for guiding the training of smaller student models through knowledge distillation.

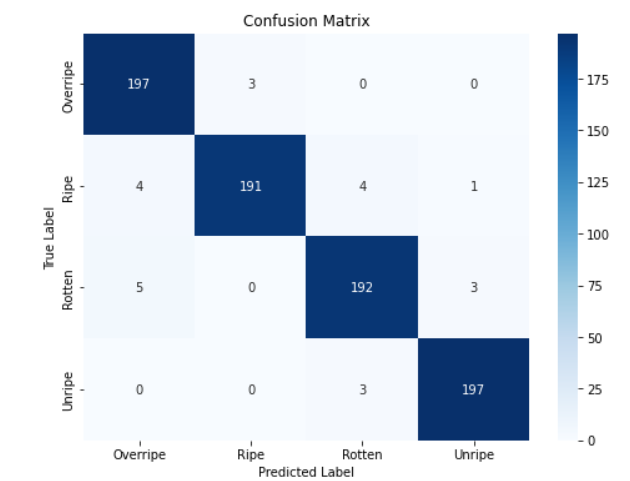
The ResNet10 student model had a 97.12% classification accuracy on the test dataset. This high accuracy demonstrates that despite having significantly fewer parameters, the distilled model effectively inherited much of the teacher’s discriminative power.

The classification report Table 2 reveals that precision, recall, and F1-scores for all four classes remain consistently high, all at or above 0.96. This shows the model’s balanced generalization across each category, without overfitting to a specific class.

Table. 2. Classification Report for ResNet10 model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-score | Support |
| Overripe | 0.96 | 0.98 | 0.97 | 200 |
| Ripe | 0.98 | 0.95 | 0.97 | 200 |
| Rotten | 0.96 | 0.96 | 0.96 | 200 |
| Unripe | 0.98 | 0.98 | 0.98 | 200 |
| Accuracy |  |  | 0.97 | 800 |
| Macro avg | 0.97 | 0.97 | 0.97 | 800 |
| Weighted avg | 0.97 | 0.97 | 0.97 | 800 |

The confusion matrix (Figure X) confirms these findings, showing that most predictions fall along the diagonal. Only a small number of misclassifications occurred, such as the three ripe bananas being classified as overripe, and five rotten bananas as overripe. These errors were minimal, showing that the student model retains strong class separation ability even with its reduced size.



**Fig. 5. Confusion matrix of the ResNet10 student model. The figure illustrates the distribution of predictions across the four ripeness categories, highlighting the effectiveness of knowledge distillation in preserving classification performance while operating with reduced computational complexity.**

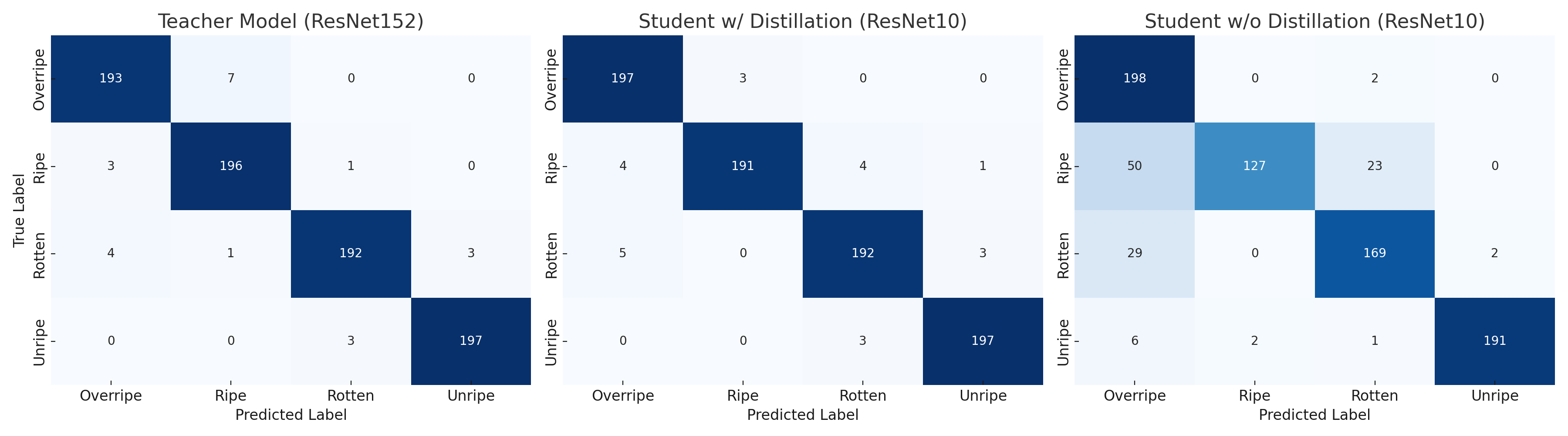
The close alignment in performance between the teacher and student models validates the effectiveness of the knowledge distillation process. It demonstrates that substantial model compression can be achieved without significant loss in accuracy, making the student model more suitable for deployment on resource-constrained environments.

To evaluate the effectiveness of the knowledge distillation, these two models were compared on the same test dataset with a student ResNet10 model with no distillation.

Table. 3. Comparison of model results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Macro Precision | Macro Recall | Macro F1-score |
| Teacher (ResNet152) | 97.75% | 0.97 | 0.97 | 0.97 |
| Student w/ Distillation (ResNet10) | 97.12% | 0.97 | 0.97 | 0.97 |
| Student w/o Distillation (ResNet10) | 85.62% | 0.89 | 0.86 | 0.85 |

The results show that the student model trained with knowledge distillation retained close to the full performance of the teacher, with an accuracy just 0.63% lower, while being much smaller and computationally efficient. But on the other hand, the same student architecture trained without distillation suffered a substantial drop in accuracy of –11.5%, especially in distinguishing between ripe and rotten categories.



**Fig. 6. Confusion matrices of the three models used in this study: ResNet152 teacher model, ResNet10 student model trained with knowledge distillation, and ResNet10 student model trained without distillation. The comparison highlights the superior performance and reduced misclassification achieved through distillation, demonstrating the effectiveness of transferring knowledge from a deeper teacher network to a lighter student network.**

The confusion matrices show that both the teacher and distilled student models achieved strong class separation. However, the student model without distillation showed several misclassifications and struggled to maintain balanced recall across all categories. These results confirm the value of knowledge distillation in transferring learned feature representations from a deep, high-capacity network to a lightweight model, enabling similar teacher accuracy while reducing computational demands for deployment on edge devices.

# **6. MODEL CONVERSION**

After the training and evaluation of the student network, the model was prepared for deployment by converting it from its standard TensorFlow SavedModel (.h5) format into the TensorFlow Lite (.tflite) format. TensorFlow Lite is a lightweight, deployment-focused framework designed for running machine learning models on resource-constrained devices such as mobile phones, single-board computers, and embedded systems (TensorFlow, n.d.). This conversion process optimizes both the model’s size and its runtime efficiency, ensuring it can be deployed in low-latency applications without requiring the full TensorFlow runtime.

The comparison between the teacher and student models highlights the benefits of this optimization. The ResNet152 teacher model occupied approximately 200 MB, making it unsuitable for many edge and mobile deployment scenarios due to high memory and storage demands. In contrast, the trained student model occupied only 7.67 MB before conversion. After conversion to TensorFlow Lite, its size was further reduced to around 7.54 MB, representing a 1.7% reduction in storage requirements from the original student version and a 96.17% reduction compared to the teacher model. This size decrease has direct implications for deployment feasibility, reducing the memory footprint, accelerating loading times, and lowering storage overhead on target devices.

The conversion process involved loading the trained student model within TensorFlow, initializing the TensorFlow Lite converter, and executing the transformation to produce the .tflite file. The resulting file retained the model’s learned parameters and architecture but in a binary format optimized for inference efficiency. Internally, TensorFlow Lite applies graph transformations and operator fusion, replacing computationally expensive operations with lighter equivalents that execute efficiently on edge hardware.

From a deployment perspective, this conversion step is essential. It transforms a high-performing but storage-heavy architecture into a portable, lightweight model that can be integrated into mobile apps, IoT devices, or embedded systems for real-time produce quality assessment. Moreover, on-device inference improves both latency and data privacy by eliminating the need for cloud-based predictions. The significant reduction in model size, while maintaining high classification accuracy, demonstrates that combining knowledge distillation with TensorFlow Lite conversion is an effective strategy for developing production-ready models.

# **7. DEPLOYMENT**

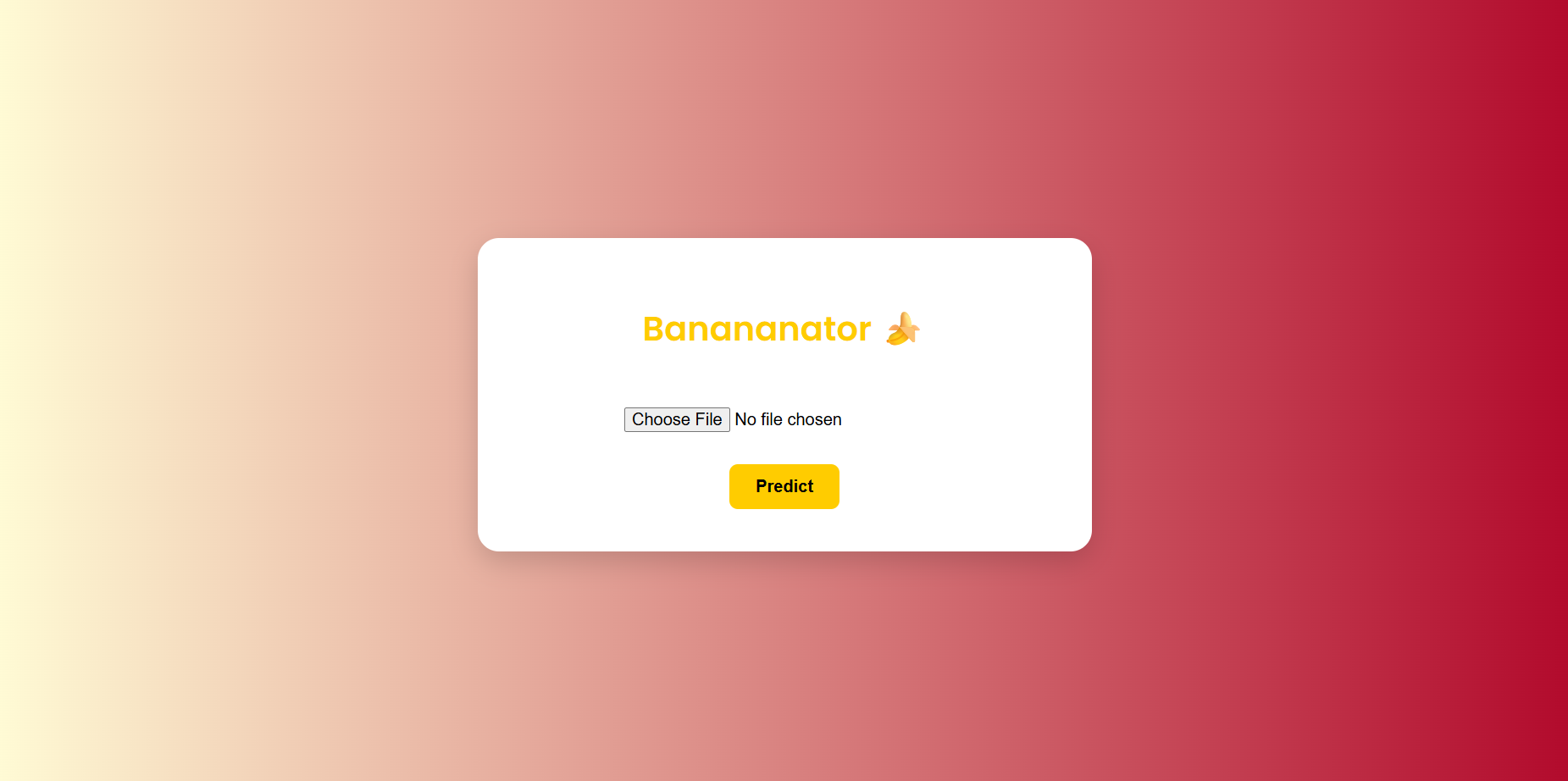
## 7.1 Web Application Interface

The trained student model, converted into TensorFlow Lite format for efficiency, was integrated into a single-page web application to provide an accessible classification tool for end-users, including retail owners, supply chain, and the general public. The front-end was built using HTML, CSS, and JavaScript, focusing on a minimalist design with intuitive interaction elements. The interface centers around a dedicated image upload field that supports drag-and-drop or manual selection, an immediate image preview function using JavaScript’s FileReader API, and a classification results panel that dynamically updates without requiring a full page reload.

When a user selects an image of a banana, the browser displays a preview alongside an interactive “Predict” button. Clicking this button triggers an HTTP POST request to the Flask backend, sending the uploaded image as part of a multipart/form-data payload. This asynchronous request allows the page to remain responsive while the server processes the classification.

On the backend, the Flask application loads the pre-optimized TensorFlow Lite model into memory using the TensorFlow Lite Interpreter API. This ensures that inference can be performed quickly without reloading the model for each request. The uploaded image is preprocessed to match the model’s expected input size of 224 × 224 pixels and normalized to a [0, 1] scale. The backend then feeds the processed image into the model for prediction, retrieves the output logits, applies argmax to determine the most probable ripeness stage, and calculates the associated confidence score as a percentage.

The prediction result, alongside the confidence score, is sent back to the front-end and rendered in a styled output panel. This immediate feedback mechanism enables users to make quick decisions on banana ripeness without navigating away from the page. Additionally, the application retains a consistent look and feel across devices through responsive CSS design, allowing it to be accessed seamlessly on mobile, tablet, and desktop screens.



**Fig. 7. Screenshot of the deployed single-page web application which allows users to upload banana images and receive real-time classification results. The interface provides prediction labels, confidence scores, and a preview of the uploaded image, demonstrating the system’s accessibility for retail owners and non-technical users.**

Originally, the deployment was planned for a Raspberry Pi with a connected camera module to facilitate real-time, in-field ripeness detection. However, hardware limitations, specifically persistent incompatibilities with the Pi camera module, prevented this approach. The shift to a web-based architecture not only eliminated these hardware constraints but also expanded accessibility, allowing users to run predictions from any internet-connected device without the need for specialized hardware.

In addition to running predictions, the backend integrates with an AWS S3 cloud storage pipeline for image archiving, storing both the uploaded banana image and its predicted ripeness label. This serves as a data lifecycle management feature, allowing future datasets to be built from real user inputs. Such an archive is valuable for continuous learning pipelines, where periodically retraining the model on fresh, real-world data can improve robustness against environmental variations. This combination of lightweight inference and cloud-based storage creates a scalable deployment framework, bridging which bridges the gap s

## 7.2 AWS Integration and Configuration

A critical enhancement to the deployment pipeline was the integration of Amazon Web Services (AWS) S3 for persistent storage of both input images and their corresponding prediction labels. This was implemented in the Flask backend using the boto3 Python SDK, which provides direct programmatic interaction with AWS services. Upon receiving an image upload and generating a ripeness prediction, the backend resets the file stream pointer and constructs a unique filename incorporating both the predicted class and a randomly generated UUID string (e.g., ripe\_3f2d7e8a.jpg). This ensures that each uploaded image is uniquely identifiable, preventing overwrites and enabling organized archiving.

The image is then uploaded to the designated S3 bucket via the upload\_fileobj() method, using credentials configured in the backend for programmatic access. The bucket was created in the eu-north-1 region, and public read access was configured to allow generated URLs to be displayed back to the user. Once uploaded, the server constructs a public S3 URL for the stored image, which is returned alongside the prediction and confidence score in the HTTP response. This enables the frontend to immediately display the stored image directly from S3, providing visual confirmation to the user.

From a research and production perspective, the S3 integration serves two strategic purposes:

1. Data Archiving: Every classified image is stored along with its predicted label, creating a growing repository of real-world data. This repository can be later reviewed, cleaned, and incorporated into extended datasets for model retraining, thereby improving the model’s generalization.
2. Scalable Storage: By offloading storage to AWS, the system avoids relying on local server storage, ensuring scalability and high availability regardless of the number of users or image volume.

This cloud integration enables the deployment to not only serve predictions but also to function as a data collection node in a continuous improvement loop, making it a foundation for future adaptive learning systems. The S3 configuration process was guided by publicly available tutorials (Sam Meech-Ward, 2021).

# **8. ETHICAL AND SOCIETAL CONSIDERATIONS**

The integration of machine learning in agriculture introduces not only technological benefits but also several ethical and societal implications that must be considered in order to ensure responsible deployment. The central motivation of this project was to support small-scale retailers and market vendors who often lack the resources to invest in advanced computational systems but still need reliable tools for assessing produce quality. However, alongside this positive contribution, concerns around food security, digital equity, data privacy, and environmental sustainability also emerge.

One of the clearest societal benefits of this system lies in its potential to reduce food waste, which remains a critical global challenge. According to the Food and Agriculture Organization of the United Nations, approximately one-third of food produced for human consumption is lost or wasted each year, contributing significantly to food insecurity and environmental degradation (FAO, 2023). Bananas are among the most wasted fruits due to their rapid ripening and short shelf life. By offering an affordable and lightweight classification tool, small-scale sellers can make more informed decisions about pricing, storage, and sales, reducing the amount of produce discarded. In this way, the project contributes directly to Sustainable Development, which emphasizes reducing food waste at both retail and consumer levels.

Despite these benefits, several ethical concerns require attention. First, this project relies on the use of a web application that integrates with Amazon Web Services (AWS) S3 for storing uploaded images and prediction results. While this provides scalability and facilitates model retraining with new data, it also introduces data privacy risks. Uploaded images, although intended to capture only bananas, may accidentally contain identifiable information such as shop environments, locations, or even people. If not carefully managed, such data could compromise user privacy. Ethical guidelines for artificial intelligence stress the importance of data minimization and anonymization to prevent misuse of personal information. Compliance with legal frameworks such as the General Data Protection Regulation (GDPR) is essential for ensuring that data handling practices are transparent, secure, and respectful of users’ rights.

Another important issue relates to fairness and bias in the dataset. The performance of deep learning models is strongly dependent on the diversity and quality of training data. If the dataset used in this project disproportionately represents bananas from specific geographic regions, the model may underperform in other contexts. For example, sellers in rural markets with different backgrounds or banana varieties may experience misclassifications, leading to reduced trust in the system. Such biases have been widely recognized as a challenge in AI applications, with calls for more inclusive and representative datasets to ensure fairness across different user groups. Incorporating continuous model updates through active learning, where newly classified images are stored in AWS S3 for periodic retraining, can help mitigate this issue by adapting the model to diverse real-world conditions.

From an environmental perspective, this project also aligns with ethical goals by reducing food waste and thereby lowering greenhouse gas emissions. Food loss is responsible for approximately 8–10% of global emissions, with bananas being a significant contributor due to their global trade volume and perishability (EPA, 2022). By helping sellers identify optimal selling windows for bananas, the system indirectly contributes to climate change mitigation. However, ethical responsibility also requires careful consideration of the system’s own computational footprint. Cloud services, while enabling scalability, are energy-intensive, and their environmental costs cannot be ignored. Ensuring efficient use of cloud resources and exploring edge-based deployment options can help balance these concerns.

Finally, there is a societal consideration around trust and transparency. AI systems deployed in real-world environments often face skepticism, particularly when decisions impact livelihoods. By adopting interpretability methods such as confidence scores alongside predictions, this project takes a step toward transparency, enabling users to better understand and evaluate the reliability of outputs. This aligns with broader ethical guidelines that emphasize explainability and accountability in AI systems (Directorate-General for Communications Networks, 2019).

In conclusion, while this project demonstrates the potential of machine learning to reduce food waste and empower small retailers, its deployment carefully considers the ethical dimensions of privacy, fairness, equity, environmental sustainability, and transparency. By proactively addressing these issues, the system maximizes societal benefit while minimizing risks, ensuring that technological innovation supports not only efficiency but also justice and sustainability.

# **9. DISCUSSIONS**

While both the teacher and distilled student models achieved high classification accuracies of 97.75% and 97.12% respectively, closer inspection through confusion matrices and classification reports reveals recurring misclassification patterns. The teacher model’s errors were relatively sparse, with occasional confusion between the Overripe and Ripe categories and between Rotten and Unripe bananas. This can be attributed to visual similarities in surface texture and color transitions during the late stages of ripening, where subtle differences may be indistinguishable even to human observers.

For the student model, although the overall accuracy remained comparable to the teacher model, there was a slight increase in cross-class confusion for the Ripe and Overripe categories. This indicates that while knowledge distillation transferred most of the teacher’s feature extraction capabilities, certain fine-grained texture and color patterns were less robustly captured due to the student model’s reduced depth and parameter count. Despite this, the model retained strong performance across all classes, with precision and recall values above 0.95 for each category.

By contrast, the non-distilled student model achieved only 85.62% accuracy, with substantial degradation in performance for the Ripe category, as seen in its confusion matrix. This model misclassified a significant number of ripe bananas as Overripe or Rotten, suggesting that direct training on the limited dataset without the teacher’s soft-label guidance led to underfitting on class-specific features. The difference between the distilled and non-distilled student models emphasizes the effectiveness of distillation in preserving generalization capabilities.

From a practical standpoint, these misclassifications could have consequences in real-world deployment, particularly in retail contexts where inaccurate classification might influence stock sorting or sales decisions. Further improvements could involve augmenting the training dataset with more borderline cases, applying targeted fine-tuning of the student model’s feature extractor, or leveraging attention mechanisms to better focus on discriminative fruit surface patterns.

# **10. LIMITATIONS OF THE STUDY**

One key limitation of this study is that the system was not tested in a real-time environment. While the web-based deployment successfully classifies uploaded images, it does not replicate continuous image capture that would occur in field settings using hardware such as a Raspberry Pi or cameras. As a result, the system’s performance under real-world remains limited. This gap highlights the need for future work to implement and evaluate the model in live operational contexts, ensuring that it performs reliably outside of controlled testing.

Also, from a technical perspective, while the ResNet10 student model is lightweight and accurate, some important information from the larger ResNet152 teacher model may have been lost during distillation. Additionally, the web-based deployment depends on internet connectivity for AWS integration, which may limit usability in low-resource areas

# **CONCLUSION**

This research explored the application of knowledge distillation for banana ripeness classification, using a high-capacity ResNet152 teacher model to train a lightweight ResNet10 student model. The approach achieved a substantial reduction in model size from the original 8 MB student model to just 2 MB after TensorFlow Lite conversion without sacrificing accuracy. Both the teacher and distilled student models achieved over 97% classification accuracy, demonstrating that a compact model can retain the performance of a deeper architecture while being more suitable for deployment on resource-constrained platforms.

The experimental results showed that the distilled student model closely matched the teacher model’s predictive capabilities, while a non-distilled student model trained directly from data underperformed significantly. This performance gap highlights the importance of transferring learned feature representations through distillation, particularly when aiming for lightweight deployment. The deployment of the final model as a single-page web application integrated with AWS S3 storage provided a practical demonstration of real-world usability, enabling efficient image classification and cloud-based archival for future model retraining.

However, despite the high accuracy, the error analysis revealed some misclassifications between visually similar ripeness stages such as Ripe and Overripe, suggesting that the model could benefit from enhanced fine-grained feature extraction. Additionally, real-world factors such as lighting variations, occlusions, and fruit orientation were not extensively addressed in the current dataset, leaving room for robustness improvements.

# **FUTURE WORK**

Future research could focus on expanding and diversifying the dataset to include a wider variety of lighting conditions, camera angles, and fruit orientations, thereby improving the model’s ability to generalize to uncontrolled real-world environments. Another promising direction is the exploration of advanced feature enhancement techniques, such as attention mechanisms or vision transformers, to enable the model to capture more subtle textural and color variations between ripeness stages.

Further work should also revisit the initial plan for Raspberry Pi deployment once the hardware limitations, particularly the camera compatibility issues, are resolved. This would enable real-time classification in agricultural fields or retail settings where portability and immediacy are crucial. Additionally, the AWS S3-based archival of classified images could be leveraged to implement an active learning framework, allowing the model to be periodically retrained with newly collected data. This would ensure that the system remains adaptive to seasonal changes and environmental variations.

Finally, the solution could be integrated into IoT-based agricultural or retail automation systems, such as automated sorting lines or inventory management platforms, to maximize operational impact. With these enhancements, the system could transition from a proof-of-concept into a scalable, production-ready tool that directly contributes to waste reduction and improved quality control in the banana supply chain.

# **ACKNOWLEDGMENTS**

I would like to sincerely thank my supervisor for his continuous guidance, constructive feedback, and encouragement throughout the development of this dissertation. I am also grateful to the faculty of the MSc in Applied Artificial Intelligence program at South East Technological University for providing the knowledge base and academic support that enabled the successful completion of this work. Also, the support of my family, friends, and colleagues has been invaluable, and I deeply appreciate their patience and encouragement during this journey.

This work also made use of generative AI tools, which were helpful in improving the clarity and readability of the report, as well as in designing and developing the web application interface. While the research design, experimentation, and analysis were conducted independently, the integration of generative AI served as a supportive aid that enhanced both the quality and efficiency of the final output.

# **REFERENCES**

Aksoy, S., Pinar Demircioglu and Ismail Bogrekci (2024). Evaluating pre-trained CNNs for distinguishing fresh vs rotten fruits and vegetables. Journal of Applied Horticulture, 26(3), pp.361–366. doi:https://doi.org/10.37855/jah.2024.v26i03.68.

Ashik, Md.A., Imtiaz, S., Mridha, Md.M.R., Shakil, Md.S.A., Nakib, Md.F.N., Khan, Md.T. and Ahammed, M.S. (2025). From Fresh to Foul: Deep Learning’s Discerning Eye on Fruit Decay. 2025 International Conference on Electrical, Computer and Communication Engineering (ECCE), [online] pp.1–6. doi:https://doi.org/10.1109/ecce64574.2025.11013966.

Baldovino, R.G., Antoine, R., Reylie, P. and Tiamzon, A.P. (2024). Real-Time Banana Ripeness Detection and Classification using YOLOv8. [online] 13, pp.219–223. doi:https://doi.org/10.1109/icom61675.2024.10652438.

Bergmann, D. (2024). Knowledge Distillation. [online] Ibm.com. Available at: https://www.ibm.com/think/topics/knowledge-distillation.

Cai, X., Zhu, Y., Liu, S., Yu, Z. and Xu, Y. (2024). FastSegFormer: A knowledge distillation-based method for real-time semantic segmentation of surface defects in navel oranges. Computers and Electronics in Agriculture, [online] 217, pp.108604–108604. doi:https://doi.org/10.1016/j.compag.2023.108604.

Chen, W.-C. and Chu, W.-T. (2020). A Study of Self Distillation for Mango Image Classification. [online] pp.356–359. doi:https://doi.org/10.1109/ics51289.2020.00077.

DETSI (2023). Fruits and vegetables that last: how to store produce properly. [online] Department of the Environment, Tourism, Science and Innovation (DETSI), Queensland. Available at: https://www.detsi.qld.gov.au/our-department/news-media/down-to-earth/fruits-and-vegetables-that-last#toc-2.

Directorate-General for Communications Networks, C. and T. (European C. (2019). Ethics guidelines for trustworthy AI. [online] Publications Office of the European Union. LU: Publications Office of the European Union. Available at: https://op.europa.eu/en/publication-detail/-/publication/d3988569-0434-11ea-8c1f-01aa75ed71a1.

Dong, Q., Gu, R., Chen, S. and Zhu, J. (2024). Apple Leaf Disease Diagnosis Based on Knowledge Distillation and Attention Mechanism. IEEE Access, 12, pp.65154–65165. doi:https://doi.org/10.1109/access.2024.3397329.

Environmental protection Agency (2022). Food waste. [online] www.epa.ie. Available at: https://www.epa.ie/our-services/monitoring--assessment/circular-economy/food-waste/.

Food and Agriculture Organization (2023). International Day of Awareness of Food Loss and Waste: FAO calls for circular model in agrifood systems. [online] Available at: https://www.fao.org/newsroom/detail/international-day-of-awareness-of-food-loss-and-waste--fao-calls-for-circular-model-in-agrifood-systems/en.

GeeksforGeeks (2025). Sparse Categorical Crossentropy vs. Categorical Crossentropy. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/deep-learning/sparse-categorical-crossentropy-vs-categorical-crossentropy/.

He, K., Zhang, X., Ren, S. and Sun, J. (2015). Deep Residual Learning for Image Recognition. [online] arXiv.org. Available at: https://arxiv.org/abs/1512.03385.

Hinton, G., Vinyals, O. and Dean, J. (2015). Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [cs, stat]. [online] Available at: https://arxiv.org/abs/1503.02531.

lonnieqin (2023). Knowledge Distillation. [online] Kaggle.com. Available at: https://www.kaggle.com/code/lonnieqin/knowledge-distillation [Accessed 18 Aug. 2025].

maimunulkjisan (2024). Knowledge Distillation using DenseNet121 Teacher. [online] Kaggle.com. Available at: https://www.kaggle.com/code/maimunulkjisan/knowledge-distillation-using-densenet121-teacher [Accessed 18 Aug. 2025].

maimunulkjisan (2024). Knowledge Distillation using InceptionV3 Teacher. [online] Kaggle.com. Available at: https://www.kaggle.com/code/maimunulkjisan/knowledge-distillation-using-inceptionv3-teacher#Data-Acquisation-and-Augmentation [Accessed 18 Aug. 2025].

Miah, M.S., Tasnuva, T., Islam, M., Keya, M., Rahman, Md.R. and Hossain, S.A. (2021). An Advanced Method of Identification Fresh and Rotten Fruits using Different Convolutional Neural Networks. 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). doi:https://doi.org/10.1109/icccnt51525.2021.9580117.

Nair, V. and Hinton, G. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. [online] Available at: https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf.

neerajmohan (2023). Model compression using knowledge distillation. [online] Kaggle.com. Available at: https://www.kaggle.com/code/neerajmohan/model-compression-using-knowledge-distillation [Accessed 18 Aug. 2025].

Pariya Afsharpour, Toktam Zoughi, Mahmood Deypir and Mohamad Javad Zoqi (2024). Robust deep learning method for fruit decay detection and plant identification: enhancing food security and quality control. Frontiers in plant science, 15. doi:https://doi.org/10.3389/fpls.2024.1366395.

ritvik1909 (2021). Knowledge Distillation. [online] Kaggle.com. Available at: https://www.kaggle.com/code/ritvik1909/knowledge-distillation [Accessed 18 Aug. 2025].

Roboflow Universe Projects (2025). Banana Ripeness Classification Dataset. [online] Roboflow. Available at: https://universe.roboflow.com/roboflow-universe-projects/banana-ripeness-classification/dataset/5.

Sam Meech-Ward (2021). Upload Images Directly to S3 from Front End. [online] YouTube. Available at: https://www.youtube.com/watch?v=yGYeYJpRWPM [Accessed 18 Aug. 2025].

Shu, Y., Zhang, J., Wang, Y. and Wei, Y. (2025). Fruit Freshness Classification and Detection Based on the ResNet-101 Network and Non-Local Attention Mechanism. Foods, [online] 14(11), p.1987. doi:https://doi.org/10.3390/foods14111987.

Speare-Cole, R. (2025). Bananas under threat as rising temperatures killing crops – report. [online] Irish Examiner. Available at: https://www.irishexaminer.com/world/arid-41630512.html.

Srivastava, C.S. (2024). ReLU vs. Sigmoid: A Comprehensive Comparison. [online] Medium. Available at: https://medium.com/%40chhayankshekhar/relu-vs-sigmoid-a-comprehensive-comparison-d8efe3f4ec04 [Accessed 18 Aug. 2025].

Stasenko, N., Savinov, M., Burlutskiy, V., Pukalchik, M. and Somov, A. (2021). Deep Learning for Postharvest Decay Prediction in Apples. IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society. doi:https://doi.org/10.1109/iecon48115.2021.9589498.

Sultana, N., Jahan, M. and Uddin, M.S. (2022). An extensive dataset for successful recognition of fresh and rotten fruits. Data in Brief, [online] 44, p.108552. doi:https://doi.org/10.1016/j.dib.2022.108552.

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, [online] 13(22), p.12504. doi:https://doi.org/10.3390/app132212504.

TensorFlow. (n.d.). Model optimization | TensorFlow Lite. [online] Available at: https://www.tensorflow.org/lite/performance/model\_optimization.

Wang, J., Wang, W., Liao, L., Luo, L., Lin, X. and Zeng, X. (2025). An Approach Based on Knowledge Distillation for Lightweight Defect Classification of Green Plums. IEEE Transactions on AgriFood Electronics, [online] pp.1–11. doi:https://doi.org/10.1109/tafe.2024.3488196.

www.image-net.org. (2017). ImageNet. [online] Available at: https://www.image-net.org/challenges/LSVRC/.

Xu, D., Dong, Y., Ma, Z., Zi, J., Xu, N., Xia, Y., Li, Z., Xu, F. and Chen, F. (2024). MAFIKD: A real-time pest detection method based on knowledge distillation. IEEE Sensors Journal, [online] pp.1–1. doi:https://doi.org/10.1109/jsen.2024.3449628.

# **Appendix 1: DATA DEFINITIONS**

Activation Function

A mathematical function applied to the output of a neural network layer to introduce non-linearity, enabling the model to learn complex patterns. Examples include Sigmoid, Tanh, and ReLU.

Amazon Web Services (AWS) S3

A cloud-based object storage service used in this study to archive banana images and classification results, enabling scalability and continuous model retraining.

Batch Normalization

A regularization technique that normalizes the activations of each layer, improving training stability, convergence speed, and reducing overfitting.

Confusion Matrix

A matrix that summarizes a classification model’s performance by showing counts of correct and incorrect predictions for each class.

Convolutional Neural Network (CNN)

A type of deep learning architecture specialized for image and visual data. It uses convolutional layers to automatically extract and learn hierarchical features.

Data Augmentation

The process of artificially expanding a dataset by applying transformations such as rotation, flipping, scaling, cropping, or color adjustments to improve generalization and robustness of models.

Dropout

A regularization method in which randomly selected neurons are ignored (dropped out) during training to prevent overfitting and improve generalization.

Edge Device

A resource-constrained device (e.g., smartphone, Raspberry Pi, IoT hardware) used for running lightweight machine learning models in real-time applications.

Evaluation Metrics

Quantitative indicators used to measure model performance. Common metrics include Accuracy, Precision, Recall, F1-score, and the Confusion Matrix.

ImageNet

A large-scale benchmark dataset containing millions of labeled images, commonly used for training and evaluating image classification models.

Inference

The process of using a trained machine learning model to make predictions on new, unseen data.

Knowledge Distillation

A model compression technique where a large, high-performing Teacher Model transfers its learned knowledge to a smaller, lightweight Student Model, preserving accuracy while reducing complexity.

Kullback–Leibler (KL) Divergence

A statistical measure of how one probability distribution differs from another. In knowledge distillation, it is used to align the probability outputs of the student with those of the teacher.

Overfitting

A condition where a model performs very well on training data but poorly on unseen data, often due to memorizing noise instead of learning general patterns.

Pooling Layer

A layer in CNNs that reduces the spatial dimensions of feature maps, typically using operations like max pooling or average pooling, to improve computational efficiency and extract dominant features.

Rectified Linear Unit (ReLU)

The most widely used activation function in deep learning. It enables faster convergence and mitigates the vanishing gradient problem.

Residual Network (ResNet)

A deep learning architecture that uses residual (skip) connections to allow gradients to flow through many layers, solving the vanishing gradient problem and enabling training of very deep networks.

Sparse Categorical Cross-Entropy (SCCE)

A loss function used for multi-class classification where labels are integer-encoded. It compares predicted probabilities against the true class index.

Teacher Model

In knowledge distillation, a large, high-capacity neural network that provides both accurate labels and “soft” probability distributions to guide the student model’s training.

Student Model

In knowledge distillation, a smaller, lightweight network trained using both the ground truth labels and the teacher model’s knowledge, designed for efficient deployment.

TensorFlow Lite (TFLite)

A lightweight version of TensorFlow designed for running optimized models on mobile devices, microcontrollers, and edge hardware with limited resources.

Vanishing Gradient Problem

A difficulty in training deep neural networks where gradients shrink exponentially as they backpropagate through layers, causing earlier layers to learn very slowly or not at all.

# **Appendix 2: TURNITIN REPORT**

