Group Mini-Project

Fruit identification and classification

Angelina Ramsunar - 41081269

Stefan Du Plooy - 40954129

Rikus Swart - 42320755

Project submitted at the North-West University

Module code: ITRI 626

Date: 12-11-2025

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The fruit quality assessment acted as the chosen scenario for the completion of this mini-project. Assessment of fruit quality remains a challenge in industries like agricultural supply chains, where classification has to happen at a fast pace with accurate results. Traditional methods of inspection work in small-scale operations but have difficulty scaling up efficiently to industrial applications. Visual assessment by human inspectors introduces variability and inconsistency, particularly when distinguishing between subtle gradations of quality. This labour intensive nature of manual sorting creates bottlenecks that may compromise both throughput and economic viability, especially in cases where time is of the essence.

Convolutional neural networks (CNNs) have emerged as a promising solution to automate fruit quality assessment through image based classification. CNNs learn hierarchical feature representations directly from pixel data, potentially capturing patterns that correlate with quality indicators such as colour, surface texture and visible features.

The task itself appears relatively simple: given labelled images categorized into discrete quality classes (Good, Mild, Rotten), a CNN should learn to distinguish among them. But this is where this project attempts to look beyond the simple objective and attempt to answer several implementation questions that arose throughout the year during classes, that remain unresolved. To what extent does colour information contribute to classification accuracy, and might grayscale representations suffice? How do different preprocessing strategies, such as data augmentation or variations in input resolution, affect model generalisation? These questions motivated our experimental design, where we systematically evaluate multiple training scenarios to understand which factors most influence performance. This is the best example of applying what has been studied as a theoretical discussion, to reflect on the practical application thereof.

Our approach involves constructing CNN architectures that process images of fruit and output quality predictions, evaluated using standard metrics such as accuracy, precision, recall, F1-score and area under the ROC curve (AUC). We implement a series of controlled experimental scenarios: baseline models with standard colour images, grayscale conversions to isolate colour dependency, augmented datasets to test robustness and multiple input. Each scenario provides insight into how specific design choices shape model behaviour, potentially revealing whether certain preprocessing steps offer marginal gains or whether simpler configurations prove sufficient for this classification task.

While the original project specification focused solely on classifying the quality of the fruit, we expanded the scope to incorporate fruit type identification as a concurrent task, thereby transforming the problem into a multi-task learning framework. This extension was motivated by two intersecting considerations.

First, the single-task formulation presented a relatively straightforward classification problem with only three output classes, offering limited opportunity to explore more interesting architectural patterns and training dynamics that exists in deep learning. By introducing an additional classification objective, identifying which specific fruit type appears in each image, we create a richer experimental context that demands more complex feature learning and allows us to investigate how shared representations can serve multiple predictive goals simultaneously.

Second, this multi-task extension addresses a topic that has come up numerous times in literature and class discussion which we attempt to investigate: the efficient utilisation of labelled data. High-quality labelled datasets is very expensive in terms of time, money and domain expertise, particularly in agricultural contexts where consistent quality standards must be maintained across diverse fruit varieties and environmental conditions. The dataset employed in this study contains images already organised by both quality level and fruit type through directory structure and filename conventions, yet limiting ourselves to quality classification alone would effectively discard half of the available label information.

The multi-task learning domain offers a mechanism to leverage this richer labelling scheme without collecting additional data, potentially improving the return on annotation investment. Recent discussions in the transfer learning literature suggest that related tasks may benefit from shared feature representations, as lower-level visual patterns such as edge detection, texture recognition and colour distribution often prove relevant across multiple classification objectives. Whether this hypothesis holds for fruit classification, where quality assessment might depend on fruit-specific characteristics, remains an empirical question we aim to address through comparative evaluation of single-task versus multi-task architectures.

The primary objective centres on developing convolutional neural network architectures capable of identifying the quality of fruit from visual data. This entails constructing models that can learn unique features directly from pixel level representations. CNNs offer particular advantages for this task through their hierarchical feature learning, where early layers may capture low-level patterns such as edges and textures, while deeper layers synthesise these into higher order abstractions that correspond to quality indicators.

But it is important to note that for this research project the objective extends beyond merely achieving acceptable classification accuracy. It involves understanding which architectural choices prove most consequential for this task and whether relatively simple networks suffice under different scenarios of input data or whether more complex designs yield proportional improvements performance.

The second objective investigates the comparative performance of single-task versus multi-task learning paradigms. Single-task models focus exclusively on quality classification, optimising their parameters to distinguish among good, mildly degraded and rotten produce. Multi-task architectures, by contrast, simultaneously predict both quality level and fruit type through a shared convolutional backbone that feeds into separate classification heads.

This dual-objective formulation raises questions about whether forcing the network to learn features useful for multiple related tasks improves generalisation or whether task interference diminishes performance on either objective. This objective speaks to broader discussions in transfer learning and multi-task optimisation about when parameter sharing proves advantageous and when task specific architectures remain preferable.

Accuracy offers an intuitive overview but may obscure class-specific performance disparities, particularly problematic when dealing with imbalanced datasets where certain quality categories appear more frequently than others. For this, various different performance metrics are introduced to analyse the output of the model and motivate the legitimacy of the findings.

Fourthly, analysing how variations in input data characteristics affect model performance represents a critical objective for understanding the practical constraints within this classification task. Colour information, for instance, might prove essential for distinguishing between early and advanced stages of degradation or texture patterns captured equally well in grayscale could suffice. Furthermore, we explore the role that image resolution plays in the trade-offs between computational efficiency and information preservation, higher resolutions retain finer details but demand greater processing resources and may increase training time substantially. Data augmentation techniques that artificially expand training sets through geometric transformations and photometric adjustments could improve robustness to variations in imaging conditions, or they might introduce artifacts that complicate learning.

The final objective synthesises findings across all experimental conditions to identify patterns that generalise beyond individual scenarios. This comparative analysis seeks to determine whether certain configurations consistently outperform alternatives regardless of specific implementation details or whether optimal approaches vary depending on contextual factors such as available computational resources, dataset characteristics, or operational requirements.

In summary, the research objectives comprise of:

1. Develop CNN-based deep learning models for fruit classification and quality assessment
2. Implement both single-task (quality only) and multi-task (fruit type and quality) learning approaches
3. Evaluate model performance using multiple metrics: accuracy, precision, recall, F1-score, ROC curves, and AUC
4. Analyse the impact of different input data characteristics on model performance
5. Compare single-task versus multi-task learning performance to assess architectural trade-offs

3.1 Dataset and model implementation

First we begin by acquiring and inspecting the dataset. The Fruit Quality Database (FruQ-DB) is used for this research study as the foundational data source. This dataset provides images of multiple fruit varieties across different quality states, organised hierarchically by fruit type and quality level.

Initial preprocessing involves structuring the data into training, validation and test partitions to support model development while maintaining strict separation between evaluation sets. The images have quality labels that are represented through directory organisation (Good, Mild, Rotten) and fruit type information are embedded in the filename prefixes. This labelling scheme enables both single-task quality classification and multi-task learning where fruit identification serves as an auxiliary objective.

For the data augmentation pipelines, we apply transformations such as random rotations, horizontal flips and adjustments to brightness and contrast, expanding the effective training set size while introducing controlled variability that may improve model robustness to more realistic unforeseen conditions not represented in the original dataset. These augmented images are saved separately, not affecting the original dataset.

For model development, we proceeded through implementation of convolutional neural network architectures in PyTorch. The architecture design follows established conventions for image classification networks, beginning with convolutional layers that extract spatial features through learned filter banks, followed by pooling operations that introduce translation invariance and reduce dimensionality. Thereafter, convolutional blocks increase filter depth while reducing spatial dimensions, a pattern that encourages hierarchical feature learning where early layers capture local patterns and deeper layers synthesise these into global representations. For single-task models, this convolutional backbone feeds into fully connected layers that produce quality class predictions. Multi-task architectures extend this design by branching after feature extraction into parallel classification heads, one for quality assessment, another for fruit type identification. This architectural choice embodies hypotheses about feature reusability across related visual tasks.

This experimental design split the research project into phases: Part A examines single-task quality classification under varying conditions, while Part B extends analysis to multi-task learning scenarios. This establishes clear baseline performance through single-task models before introducing the additional complexity of multi-task optimisation, allowing direct assessment of whether auxiliary objectives provide benefits or merely introduce confounding factors.

3.2 The two research parts and different scenarios

Part A comprises three scenarios that manipulate input characteristics while maintaining focus on quality classification alone. Scenario 1 establishes baseline performance using standard RGB images pre-processed through resize operations and normalisation. This acts as an unaltered baseline for future comparisons that change the input scenarios. Scenario 2 converts images to grayscale, testing the hypothesis that colour information contributes critically to quality assessment, or conversely, that texture and structural features captured in intensity values is good enough for accurate classifications. This scenario addresses practical considerations about whether simpler single-channel representations might reduce computational requirements without sacrificing performance. Scenario 3 introduces data augmentation during training, applying random transformations that expand the effective dataset size while potentially improving generalisation to images captured under conditions not represented in the original training distribution. The dataset provides near perfect images, this is why Scenario 3 is so important to test the model to more realistic variation of image inputs.

Part B replicates each of these scenarios within a multi-task learning framework where models simultaneously predict fruit type and quality level. Scenario 1 establishes multi-task baseline performance with standard RGB inputs. Scenario 2 examines whether grayscale conversion affects both tasks equally or whether fruit type identification proves more colour dependent than quality assessment. Scenario 3 evaluates augmentation under multi-task learning, questioning whether synthetic variations benefit both objectives or introduce task specific biases that help one classification head while punishing or degrading the other.

3.3 Training and Results

A fixed random seed is used to control to ensure that results are repeatable and reproducible. Train/validation/test splits maintain strict separation, with test sets reserved exclusively for final evaluation after all architectural decisions and hyperparameter selections are complete. Each scenario generates comprehensive logs capturing training dynamics, validation metrics across epochs and final test set performance.

Results analysis synthesises findings across all scenarios through multiple analytical lenses. Quantitative comparison of performance metrics identifies which configurations achieve superior classification accuracy, precision, recall, and F1-scores, while ROC curves and AUC values reveal discriminative capacity across varying decision thresholds. Confusion matrices expose class-specific error patterns, indicating whether models systematically misclassify particular quality levels or fruit types, which represents insights that might suggest targeted refinements or reveal inherent ambiguities in the classification task itself. Then finally cross scenario comparisons assess the impact of individual factors: does grayscale conversion consistently degrade performance by some quantifiable margin, or do effects vary depending on whether models operate in single-task or multi-task mode? Do augmentation benefits depend on resolution, suggesting interactions between preprocessing choices?

Training curves track loss and accuracy evolution across epochs indicate whether models converge reliably or exhibit instability, whether they overfit to training data or maintain consistent performance across training and validation sets.

CONVOLUTIONAL NEURAL NETWORK

As previously mentioned, this research project makes use of the Fruit Quality Database (FruQ-multi), which is a comprehensive image dataset specifically curated for assessing the produce quality across multiple fruit and vegetable varieties. The dataset comprises 9370 images spanning 11 distinct fruit types, each annotated according to quality level.

The dataset encompasses eleven fruit and vegetable categories. Within each fruit category, images are further subdivided into three quality classes that represent progressive stages of degradation. The “Good” or “Fresh” category contains images of high quality produce exhibiting minimal surface defects and absence of visible decay. The “Mild” category represents moderate quality with minor blemishes, slight discoloration or early indicators of degradation but do not render the produce unsuitable. The “Rotten” category encompasses poor quality fruits displaying extensive bruising, mold growth or structural collapse that clearly indicate spoilage.

The dataset distribution across fruit types and quality classes reveals substantial heterogeneity. Table 1 presents the complete breakdown of image counts. It is important to take note of the imbalances both between fruit types and within quality categories, as this will have an effect on the later stages of using the data. Tomatoes constitute the most represented fruit with 1990 images, while strawberries comprise the smallest subset with only 216 images. Quality distribution within individual fruits also exhibits considerable variation. The PepperQ subset, for instance, contains 660 rotten images but only 48 good images. Conversely, PearQ maintains relatively balanced representation across good (504), mild (493), and rotten (100) categories. Notably, the StrawberryQ subset entirely lacks a “Good” class, containing only mild (119) and rotten (97) images.

Fruit Type	Good/Fresh	Mild	Rotten	Total
BananaDB	179	96	337	612
CucumberQ	250	345	116	711
GrapeQ	227	194	288	709
KakiQ	545	226	340	1111
PapayaQ	130	250	413	793
PeachQ	425	136	584	1145

PearQ	504	493	100	1097
PepperQ	48	24	660	732
StrawberryQ	0	119	97	216
tomatoQ	600	440	950	1990
WatermeloQ	51	53	150	254
Total	**3009**	**2376**	**3985**	**9370**

Table 1: Distribution of images across fruit types and quality classes in the FruQ-multi dataset

Class imbalance, specifically in the subsets of PepperQ and StrawberryQ, has the risk of creating unwanted bias in models to over predict the majority classes if training is not adapted appropriately. This means a model can just predict the most common class for every input and will have a high accuracy in the end. The absence of certain quality categories in specific fruit types complicates multi-task learning where models must simultaneously predict fruit identity and quality. For example training on strawberries inherently provides no gradient signal for the “Good” category, potentially degrading the shared representation's ability to encode features associated with high quality.

The dataset size variations across fruit types also affects learning dynamics. Tomatoes contribute for over 20% of all images, potentially dominating learned features if the model inadvertently specialises for this overrepresented category. These considerations informed preprocessing decisions that are discussed in the subsequent sections.

The data was partitioned by using stratification to split it, to maintain proportional representation of quality classes across training, validation and test sets. The training partition, comprising approximately 60-70% of available images, provides the primary learning source from which models extract feature representations and optimise classification boundaries. The validation set, constituting roughly 15-20% of data, serves dual purposes during model development. It is used to monitor training progress, to detect overfitting and informing hyperparameter selection. Critically, validation data influences model selection without directly participating in gradient updates, providing unbiased estimates of generalisation performance during iterative refinement. The test set, also approximately 15-20% of images, remains isolated until final evaluation, ensuring that reported performance metrics reflect genuine generalisation to unseen data. This three way partition aligns with established machine learning practice for maintaining independence between model development and final assessment.

Not all the images in the dataset are the same size. This is compensated for by standardising all the input before it is fed to the model. Original images arrive in PNG format with three colour channels (RGB), preserving spatial information at resolutions typically exceeding 224×224 pixels. Preprocessing pipelines normalise pixel intensities using ImageNet statistics, a conventional practice that stabilises gradient magnitudes during backpropagation and potentially improves convergence rates. Where scenario specific transformations occur like with grayscale conversion, augmentation or resolution adjustment, we apply the modifications on top of the baseline preprocessing. This ensures all inputs are standardised by the data preprocessing pipeline and then adapted for each scenario to ensure all models start with exactly the same input data.

The original FruQ-multi dataset arrives organised hierarchically by fruit type, with quality subdivisions nested within each fruit category. Initial preprocessing begins with restructuring operations that merge fruit specific directories into quality based partitions, creating unified training, validation and test sets that contains random examples and samples of each fruit class. Within this dataset there are also naming inconsistencies. Some folders are named “Fresh” while others are named “Good” and there are also instances of “mild” versus “Mild”. These naming inconsistencies must also be addressed and fixed during the preprocessing phase. The reorganisation employed a custom Python script that iterates through source directories, normalises inconsistent quality labels and redistributes images according to stratified sampling to maintain proportional class representation across splits.

Filename conventions established during reorganisation preserve fruit type information through systematic prefixes appended to original filenames. An image originally located at “BananaDB/Good/image001.png” transforms into “Good/BananaDB_image001.png” within the restructured training directory, encoding both quality and fruit identity in a format accessible to automated label extraction. The consistent application of this naming convention across all 9370 images facilitates programmatic label parsing through string manipulation operations that split filenames on underscore delimiters and map prefixes to integer class indices during data loading.

The resize operations standardise the different native dimensions present in the original dataset into uniform 224×224 pixel representations.

Through the operation of batch construction, the individual pre-processed images are assembled into mini batches of 32. The chosen batch size of 32 represents conventional practice in computer vision applications where memory constraints often preclude substantially larger batches given the spatial dimensionality of image tensors, a single 224×224 RGB image in 32-bit floating point format requires approximately 600KB, so a batch of 32 images consumes roughly 19MB before accounting for intermediate activations that multiply memory requirements during forward and backward passes. This is why a batch size of 32 remains optimal under varying computational availability in the group.

For scenario 3, validation and test set preprocessing deliberately excludes augmentation operations applied during training, maintaining identical transformations across both partitions. This consistency proves critical for obtaining unbiased performance estimates, in applying different preprocessing to evaluation data than the model encountered during training would conflate two effects: genuine generalisation performance on held out examples versus sensitivity to preprocessing differences. The absence of augmentation in validation and test sets also reflects deployment scenarios where incoming images arrive without opportunities for online augmentation, making evaluation on unaugmented data more representative of real world performance.

The convolutional neural network architecture used in this research follows hierarchical feature extraction, beginning with pixel level representations and progressively synthesising these into abstract feature vectors suitable for classification decisions.

Input images enter the network as three dimensional tensors with shape (channels, height, width), specifically, (3, 224, 224) for RGB inputs or (1, 224, 224) for grayscale variants, where each pixel value resides in the range [0, 1] after initial rescaling and then centres near zero following normalisation with ImageNet statistics. This numerical representation transforms photographic images into matrices of floating point values that can be used by convolutional operations.

The primary architecture, implements a four block convolutional design where each block comprises a convolutional layer, batch normalisation, ReLU activation and max pooling operations that are sequentially executed.

The first convolutional block processes input channels (three for RGB, one for grayscale) through 32 learned 3×3 filters, producing 32 feature maps that has the goal of capturing low level patterns such as edges, corners and simple textures. Batch normalisation follows, standardising activations across the mini-batch to stabilise training dynamics. The ReLU activation introduces nonlinearity by zeroing negative values while preserving positive activations unchanged. Max pooling with 2×2 windows and stride of 2 reduces spatial dimensions by half along each axis, selecting maximum activation values within local neighbourhoods.

The subsequent convolutional blocks follow this pattern with increasing filter counts: block two employs 64 filters, block three uses 128 and block four applies 256, doubling channel depth at each stage while halving spatial dimensions. After four pooling operations, the 224×224 input reduces to 14×14 spatial dimensions ($224 / 2^4 = 14$), and with 256 channels, the resulting feature tensor contains 14×14×256 = 50176 elements that encode hierarchical visual information extracted through the convolutional pipeline. These spatial features undergo flattening into a single-dimensional vector before entering fully connected layers that perform final classification decisions.

The classification head comprises three fully connected layers that progressively reduce dimensionality from 50176 input features through intermediate representations of 512 and 256 dimensions before producing final logits for the three quality classes (Good, Mild, Rotten) or fourteen output dimensions for multi-task scenarios (the three quality classes plus eleven fruit types).

Dropout layers with probability 0.5 appear between fully connected layers during training, randomly zeroing half of the activations. The network outputs raw logits rather than probabilities; during inference, these pass through a SoftMax function that exponentiates and normalises values to produce probability distributions over classes, though training employs CrossEntropyLoss that combines SoftMax and negative log likelihood for numerical stability.

Layer	Operation	Output Shape	Parameters
Input	Normalised pixel values [−2, 2]	3 × 224 × 224	—
Conv Block 1	Conv2d (3→32, 3×3) + BN + ReLU + MaxPool(2×2)	32 × 112 × 112	928
Conv Block 2	Conv2d (32→64, 3×3) + BN + ReLU + MaxPool(2×2)	64 × 56 × 56	18560
Conv Block 3	Conv2d (64→128, 3×3) + BN + ReLU + MaxPool(2×2)	128 × 28 × 28	73984
Conv Block 4	Conv2d (128→256, 3×3) + BN + ReLU + MaxPool(2×2)	256 × 14 × 14	295424
Flatten	Reshape spatial dimensions	50,176	—
FC Layer 1	Linear (50,176→512) + ReLU + Dropout(0.5)	512	25690624
FC Layer 2	Linear (512→256) + ReLU + Dropout(0.5)	256	131328
Output Layer	Linear (256→3) [Quality classes]	3	771

Total Parameters	**26211619**

Table 2: SimpleCNN Architecture: Hierarchical feature extraction through convolutional blocks

Multi-task architectures extend this base design by branching after the shared convolutional backbone into parallel classification heads. Rather than a single output layer predicting quality classes, multi-task models maintain the shared feature extractor (all four convolutional blocks plus the first fully connected layer) but split into two task specific pathways: one head containing a 256-unit hidden layer followed by three quality class outputs, another with identical structure but eleven fruit type outputs.

Table 3 summarises all configurable parameters employed across the scenarios, presenting default values alongside permissible ranges of alternative options. The distinction between single-task and multi-task parameters highlights architectural differences where multi-task configurations introduce additional variables (such as class counts for dual heads, loss weights for task balancing) that have no analogue in single-objective learning scenarios.

Parameter Category	Parameter	Default Value	Options/Range
Data Parameters	IMG_SIZE	224	64, 128, 224, 299
	BATCH_SIZE	32	8, 16, 32, 64
	NUM_WORKERS	4	0-8 (CPU threads)
	GRAYSCALE	False	True, False
	AUGMENT	False	True, False
Model Parameters	MODEL_NAME	simple	simple, deep, light
	INPUT_CHANNELS	3	1 (grayscale), 3 (RGB)
Training Parameters	NUM_EPOCHS	50	10-200
	LEARNING_RATE	0.001	0.0001-0.01
	WEIGHT_DECAY	1e-4	0-0.01
	OPTIMIZER	adam	adam, sgd, adamw
	SCHEDULER	plateau	plateau, cosine, step

	PATIENCE	10	5-20 epochs
Advanced Training	WARMUP_EPOCHS	5	0-10
	GRAD_CLIP	1.0	0.1-5.0, None
	MIXED_PRECISION	True	True, False
Multi-Task Parameters	NUM_QUALITY_CLASSES	3	Fixed: Good, Mild, Rotten
	NUM_FRUIT_CLASSES	11	Fixed: 11 fruit types
	QUALITY_LOSS_WEIGHT	1.0	0.1-2.0
	FRUIT_LOSS_WEIGHT	1.0	0.1-2.0
Other Parameters	USE_CLASS_WEIGHTS	True	True, False
	SEED	42	Any integer

Table 3: Summary of configurable parameters and their values across experimental scenarios

PART A

8.1 Introduction

Scenario 1 establishes the baseline performance for future reference in Part A of this research project using standard RGB colour images. This scenario processes unmodified three-channel colour images at 224×224 resolution, providing the foundation against which subsequent experimental modifications can be compared.

8.2 Configuration

Table 4 presents the complete configuration for Scenario 1. The configuration emphasises reproducibility through fixed random seeding (seed = 42. The model processes standard three-channel RGB images without augmentation or preprocessing beyond resizing and normalisation.

Parameter	Value
Model Architecture	Simple CNN
Input Channels	3 (RGB)
Image Size	224 × 224
Batch Size	32
Epochs	50
Learning Rate	0.001
Optimizer	Adam
Weight Decay	0.0001
Scheduler	ReduceLROnPlateau
Patience	10 epochs
Gradient Clipping	1.0
Warmup Epochs	5

Mixed Precision	Enabled
Class Weights	Enabled (auto-computed)
Augmentation	Disabled
Grayscale	Disabled
Random Seed	42

Table 4: Configuration parameters for Scenario 1 baseline

8.3 Results and Analysis

8.3.1 Overall Performance Metrics

The baseline RGB model achieved excellent performance across all evaluation metrics, demonstrating the effectiveness of CNN-based approaches for fruit quality assessment. Table 2 presents the comprehensive performance metrics on both validation and test sets, revealing consistent high-accuracy classification with minimal error rates.

Metric	**Validation**	**Test**
Accuracy	99.84%	99.79%
Precision	99.84%	99.79%
Recall	99.84%	99.79%
F1-Score	99.84%	99.79%
AUC	1.0000	0.9998
Total Errors	3 / 1,872	2 / 939

Table 5: Overall quality classification performance for RGB baseline

The validation set achieved 99.84% accuracy with only 3 misclassifications among 1872 samples, while the test set achieved 99.79% accuracy with 2 errors among 939 samples. The near-perfect AUC scores (1.0000 for validation, 0.9998 for test) indicate excellent discriminative capability across all decision thresholds. The consistency between validation and test performance (difference of 0.05 percentage points) demonstrates robust generalisation without overfitting, suggesting that the model learned genuine quality assessment patterns rather than memorising training data.

The minimal error rates (0.16% on validation, 0.21% on test) indicate that RGB colour images provide rich discriminative information for fruit quality classification. The model successfully leverages colour features, texture patterns and shape characteristics to distinguish between Good, Mild and Rotten fruit with high reliability. These results establish a strong baseline for subsequent experimental scenarios exploring alternative input representations or training strategies.

8.3.2 Per Class Performance Analysis

Table 3 presents all three quality classes achieved performance exceeding 99.3% across all metrics, demonstrating balanced classification capability.

Class	Split	Precision	Recall	F1-Score
Good	Validation	100.00%	100.00%	100.00%
	Test	100.00%	99.66%	99.83%
Mild	Validation	100.00%	99.37%	99.68%
	Test	99.17%	100.00%	99.58%
Rotten	Validation	99.63%	100.00%	99.81%
	Test	100.00%	99.75%	99.88%

Table 6: : Per class quality classification performance for RGB baseline

The "Good" class achieved perfect performance on the validation set (100% across all metrics) and near-perfect performance on the test set (99.66% recall). The "Mild" class showed 99.37% recall on validation and perfect 100% recall on test, with precision values exceeding 99%. The "Rotten" class achieved perfect recall (100%) on validation and 99.75% on test, with precision of 99.63% and 100% respectively.

The balanced performance across quality categories indicates that the model does not exhibit systematic bias towards any particular class. All three quality levels are classified with high reliability, suggesting that the RGB features effectively capture discriminative patterns for each quality category.

8.4 Confusion Matrix Analysis

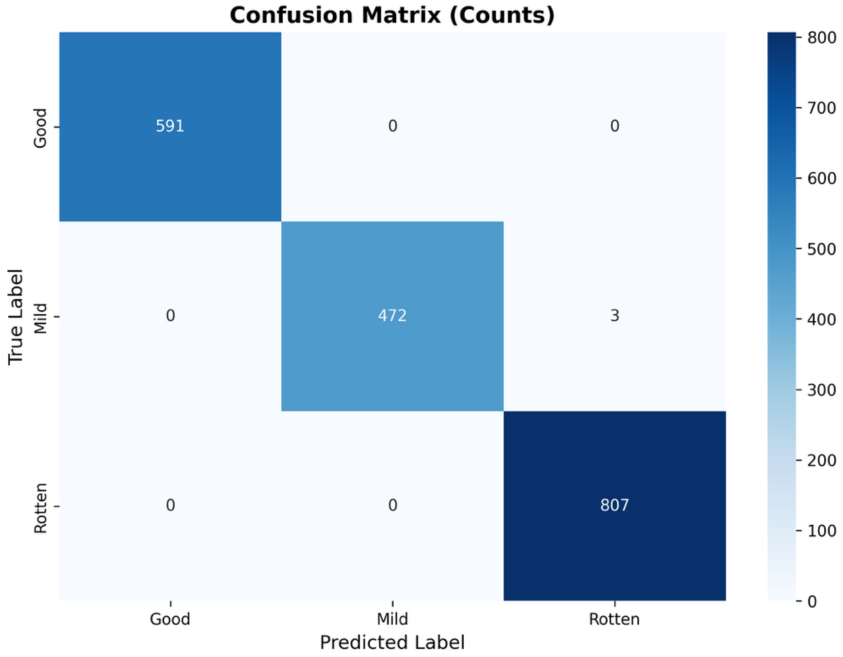

Figure 8-1: Validation Set Confusion Matrix

The validation confusion matrix reveals three misclassifications among 1872 samples. All three errors involved "Mild" samples being classified as "Rotten," indicating that the model encountered difficulty distinguishing between intermediate-quality fruit showing early deterioration and severely degraded fruit. Critically, the confusion matrix shows perfect discrimination at the quality extremes: no "Good" samples were misclassified as "Rotten" or vice versa, and no "Good" samples were confused with "Mild." The "Rotten" class achieved perfect classification with zero errors.

This error pattern suggests that classification difficulty occurs specifically at the boundary between Mild and Rotten quality categories, where visual features may be ambiguous. Fruit in transitional degradation states may exhibit characteristics of both categories, making definitive classification challenging even with full colour information.

8.5 ROC Curve Analysis and AUC Scores

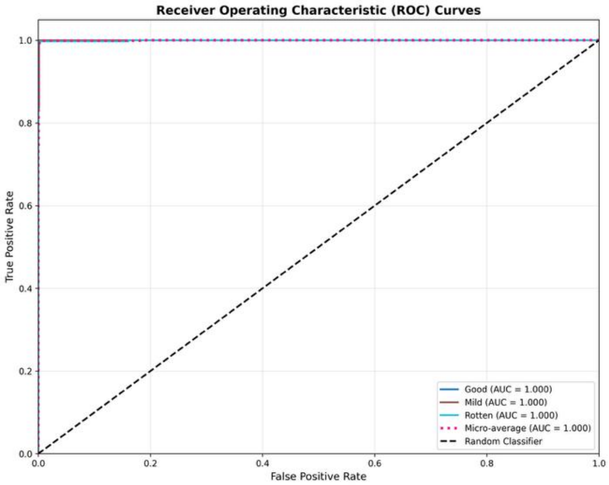

Figure-8-2: ROC test

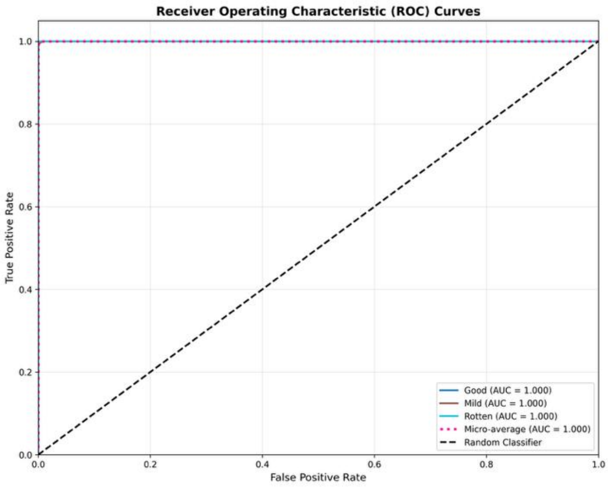

Figure-8-3: ROC validation

The ROC curves demonstrate exceptional discriminative capability for RGB-based quality classification. All three quality classes achieved perfect or near-perfect AUC scores (Table 5): validation AUC = 1.0000 for all classes with micro-average AUC = 1.0000; test AUC = 1.0000 for all classes with micro-average AUC = 1.0000. These perfect AUC scores indicate that the model achieves optimal rank-ordering of predictions across all probability thresholds.

The ROC curves for all classes track along the upper-left corner (point [0,1]), indicating that the model achieves maximum true positive rate while maintaining minimal false positive rate across all decision thresholds. This ideal behaviour confirms that the model's predicted probabilities are well calibrated, with clear separation between correct and incorrect class predictions. When the model assigns high confidence to a prediction, that prediction is correct with extremely high probability.

8.6 Training History and Convergence Analysis

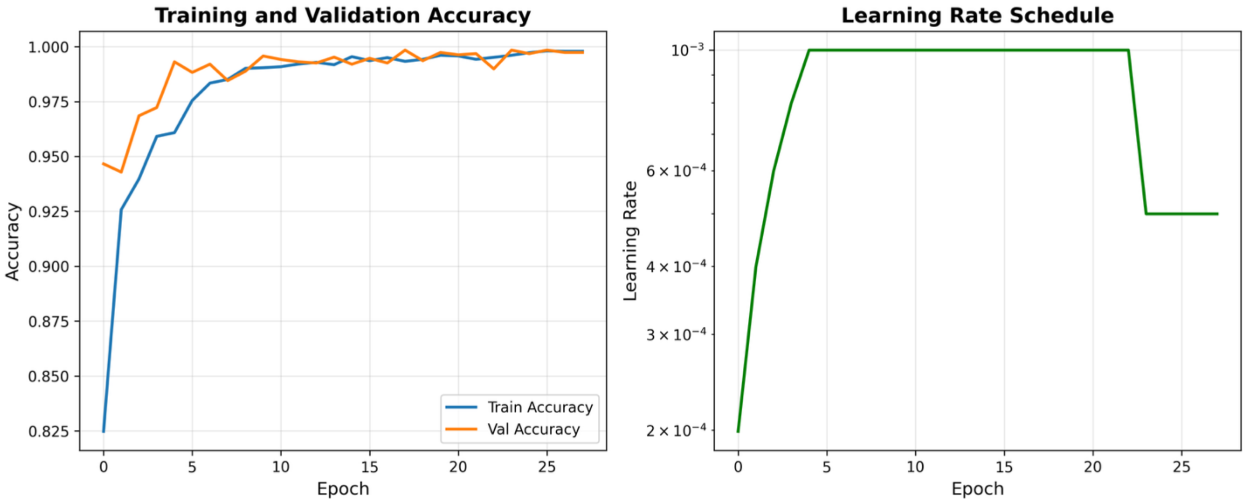

Figure 8-4: Training history

The training history provides critical insight into the model's learning dynamics and confirms robust generalisation without overfitting. Analysis of loss curves, accuracy trajectories and learning rate adjustments reveals stable convergence with consistent validation performance.

8.6.1 Convergence Dynamics

The model converged rapidly within the first 10 epochs, achieving >99% accuracy by epoch 5 and stabilising at >99.5% by epoch 8. Both training and validation accuracy curves tracked each other closely throughout training, with validation accuracy occasionally matching or slightly exceeding training accuracy. This pattern indicates genuine learning rather than memorisation.

8.6.2 Loss Reduction Dynamics

The training loss decreased from approximately 0.48 to near-zero by epoch 10, while validation loss decreased from 0.16 to near-zero following a similar trajectory. The lower initial validation loss (0.16 vs. 0.48 training) likely reflects differences in batch normalisation behaviour between training and evaluation modes, rather than indicating that the validation set is easier to classify. Both loss curves show a smooth decrease without oscillations, indicating stable gradient descent dynamics and appropriate learning rate selection.

8.6.3 Learning Rate Schedule Impact

The learning rate schedule shows two reduction events triggered by the ReduceLROnPlateau scheduler when validation loss plateaued. The initial learning rate of 0.001 was maintained through the 5-epoch warmup phase and early convergence (epochs 1-15). The first reduction occurred around epoch 15, dropping the learning rate to approximately 5 × 10^{-4}. A second reduction occurred around epoch 28, further decreasing to approximately 2.5 × 10^{-4}.

8.7 Conclusion

Scenario 1 establishes a robust baseline for fruit quality assessment using standard RGB colour images, achieving 99.84% validation accuracy and 99.79% test accuracy. The model successfully captures discriminative visual features from colour images, including colour shifts, texture patterns and surface characteristics that indicate fruit quality.

The systematic evaluation methodology applied in this baseline scenario by looking at per class metrics, confusion matrix analysis, ROC curve examination, and training history inspection, provides a comprehensive framework for assessing model performance beyond simple accuracy metrics. This multi-faceted approach reveals not only how well the model performs but also why it performs well, building confidence in the model's reliability for practical fruit quality assessment applications. Future scenarios can use this baseline as a reference point for evaluating the impact of experimental modifications on classification performance.

9.1 Introduction

In scenario 2 we explore the role of colour information images by converting all input images to grayscale. This input manipulation addresses a fundamental: is colour information essential for accurate classification (in fruit quality classification in this case), or can texture and shape features alone achieve comparable performance? By maintaining identical model architecture, training procedures and hyperparameters while only modifying the input representation from RGB (3 channels) to grayscale (1 channel), this scenario provides a controlled comparison to establish the necessity of colour features.

9.2 Configuration

Table 1 presents the configuration for Scenario 2. The configuration differs from Scenario 1 in only two parameters: INPUT_CHANNELS (reduced from 3 to 1) and GRAYSCALE (changed from false to true). All other parameters remain identical to ensure a fair comparison.

Parameter	Value
Model Architecture	Simple CNN
Input Channels	***1 (Grayscale)***

Image Size	224 × 224
Batch Size	32
Epochs	50
Learning Rate	0.001
Optimizer	Adam
Weight Decay	0.0001
Scheduler	ReduceLROnPlateau
Patience	10 epochs
Gradient Clipping	1.0
Warmup Epochs	5
Mixed Precision	Enabled
Class Weights	Enabled (auto-computed)
Augmentation	Disabled
Grayscale	***Enabled***
Random Seed	42

Table 7: Configuration parameters for Scenario 2

9.3 Results and Analysis

9.3.1 Overall Performance Metrics

Scenario 2 achieved remarkable performance that challenges conventional assumptions about the necessity of colour information for fruit quality assessment. Table 8 presents the comprehensive performance metrics, revealing that grayscale images not only maintained high accuracy but actually achieved perfect performance on the test set.

Metric	Validation	Test	Δ from S1

Accuracy	99.84%	100.00%	+0.21%
Precision	99.84%	100.00%	+0.21%
Recall	99.84%	100.00%	+0.21%
F1-Score	99.84%	100.00%	+0.21%
AUC	0.9994	1.0000	+0.0002

Table 8: Overall quality classification performance

The test set results are particularly striking: 100% accuracy across all metrics, representing a perfect classification of all 939 test samples. This performance actually surpasses Scenario 1's already exceptional 99.79 % test accuracy by 0.21 percentage points. The validation set achieved 99.84% accuracy, marginally lower than Scenario 1 (99.89%) but still indicating excellent generalisation. The slight performance difference between validation (99.84%) and test (100%) suggests some stochastic variation but no evidence of overfitting.

9.4 Per Class Performance Analysis

Table 9 presents detailed per-class metrics, revealing how grayscale conversion affected classification performance across different quality categories.

Class	Split	Precision	Recall	F1-Score
Good	Validation	100.00%	99.83%	99.92%
	Test	100.00%	100.00%	100.00%
Mild	Validation	99.79%	99.58%	99.68%
	Test	100.00%	100.00%	100.00%
Rotten	Validation	99.75%	100.00%	99.88%
	Test	100.00%	100.00%	100.00%

Table 9: Per class quality classification performance for grayscale images

The test set achieved perfect 100% metrics across all three quality classes, indicating that grayscale features were sufficient to distinguish Good, Mild, and Rotten fruit without any classification errors. On the validation set, the "Mild" class showed the most variability with 99.58% recall representing one additional misclassification. The "Good" class maintained identical recall while the "Rotten" class achieved perfect recall (100%). These results demonstrate that the intermediate "Mild" quality category is marginally more challenging without colour information, though the difference is minimal (0.21 percentage points).

9.5 Confusion Matrix Analysis

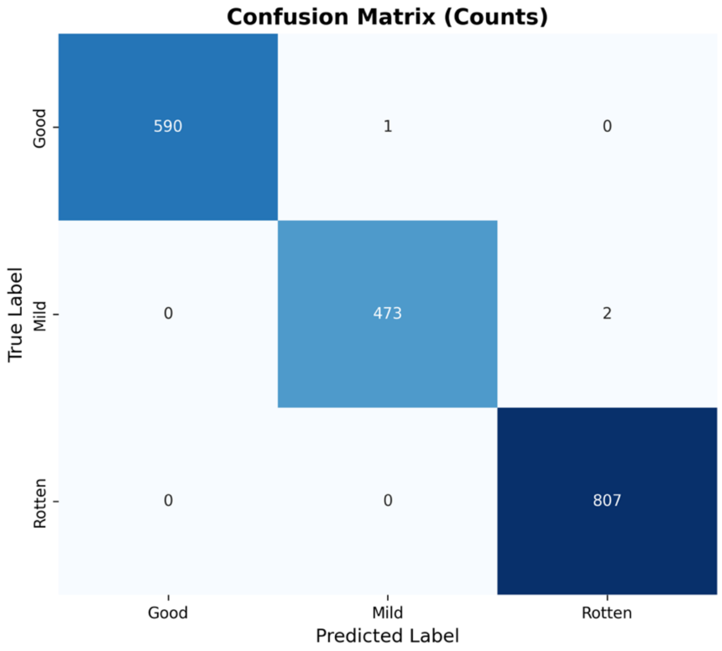

Figure 9-1: Validation confusion matrix

Figure 9-1 reveals three misclassifications among 1872 samples: one "Good" sample classified as "Mild," and two "Mild" samples classified as "Rotten." Scenario 2 introduced some additional errors, specifically an extra "Mild" versus "Rotten" misclassification. This pattern suggests that without colour information, the model found it slightly more difficult to distinguish between mid-grade fruit showing early deterioration (Mild) and severely deteriorated fruit (Rotten). However, the model maintained perfect discrimination at the extremes: no "Good" samples were misclassified as "Rotten" and vice versa.

9.6 ROC Curve Analysis and AUC Scores

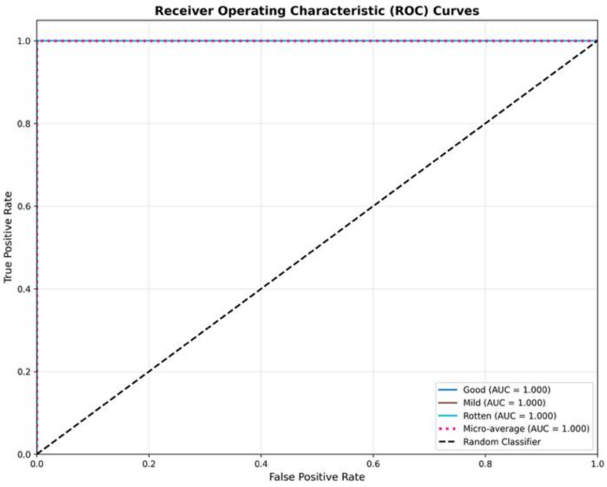

Figure 9-2: ROC test

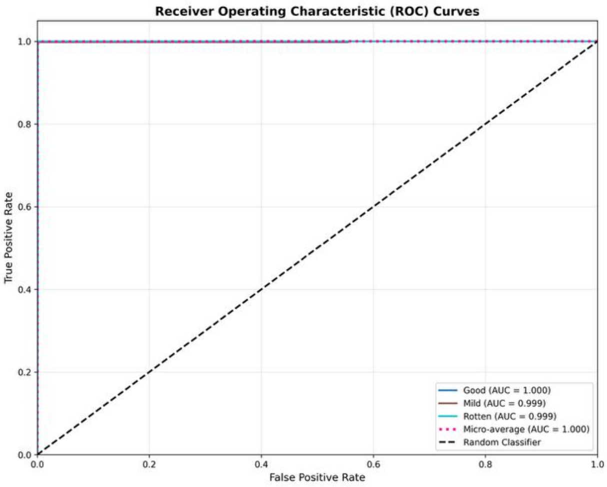

Figure 9-3: ROC validation

Figure 9-2 and Figure 9-2 demonstrate exceptional discriminative capability for grayscale-based classification. The validation set achieved near-perfect AUC scores (Table 8): 1.0000 for "Good," 0.9990 for "Mild," and 0.9990 for "Rotten," with a micro-averaged AUC of 1.0000. The test set achieved perfect 1.0000 AUC for all classes and micro-average. These perfect AUC scores indicate that the model achieves optimal rank-ordering of predictions across all probability thresholds.

The validation ROC curves for "Mild" and "Rotten" classes show slight departure from the perfect upper-left corner, with curves passing through approximately (0.002, 0.995) and (0.001, 0.998) respectively before reaching (0, 1). This minimal deviation indicates that the model occasionally assigned slightly lower confidence to a few correct predictions, but maintained near-perfect rank ordering overall. The "Good" class maintained a perfect ROC curve even on validation data, suggesting that high-quality fruit exhibits distinctive grayscale features that are easily distinguished from degraded fruit. The test set ROC curves track the upper-left corner perfectly for all classes, consistent with the 100% accuracy achieved.

9.7 Training History and Convergence Analysis

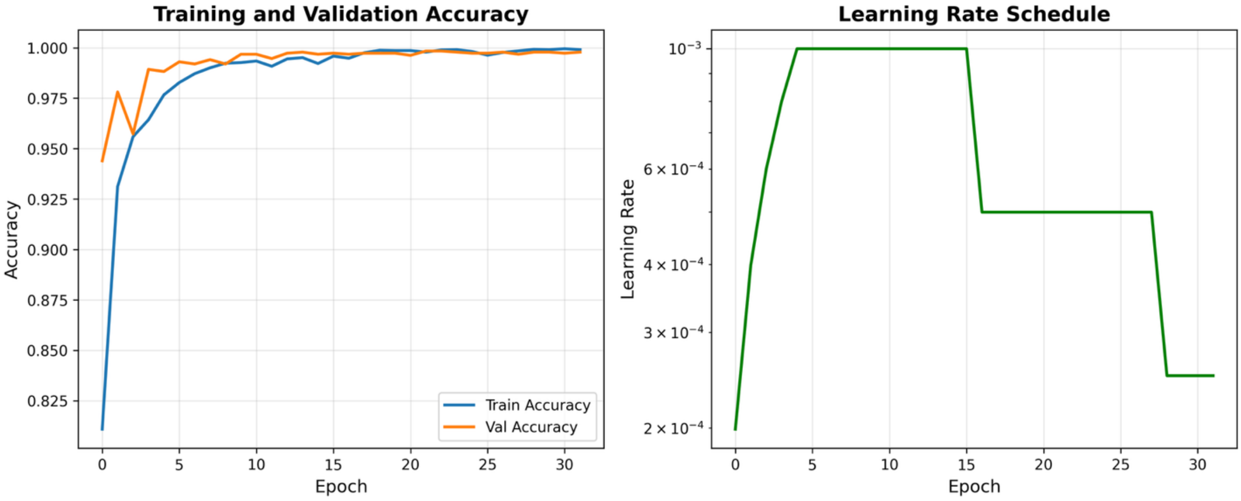

Figure 9-4: Training history

Figure 9-4 reveals how the grayscale-based model learned quality discrimination patterns and provides evidence for robust generalisation without overfitting.

9.7.1 Convergence Dynamics

The model converged rapidly within the first 10 epochs, achieving >99% accuracy by epoch 5 and stabilising near 100% by epoch 10. Both training and validation accuracy curves tracked each other closely throughout training, with validation accuracy slightly leading training accuracy in several epochs, a pattern opposite to overfitting. The training loss decreased from approximately 0.48 to near-zero by epoch 10, while validation loss similarly decreased from 0.14 to near-zero.

9.7.2 Learning Rate Schedule Impact

The learning rate schedule shows two reduction events triggered by the ReduceLROnPlateau scheduler. The initial learning rate of 0.001 was maintained through the warmup phase and early convergence. The first reduction occurred around epoch 15, dropping the learning rate to approximately 5×10^{-4}, followed by a second reduction around epoch 28 to approximately 2.5×10^{-4}. These adaptive reductions enabled progressive refinement of decision boundaries.

9.7.3 Evidence Against Overfitting

Traditionally when models output near perfect results, this is an immediate red flag to look for overfitting. But given the context of this project, where we have near perfect images in our dataset that are high definition and a very simple classification problem with only three categories, we can expect near perfect outputs. This is not enough to justify the absence of overfitting, thus looking at the following statements derived from the outputs we can debate the argument that this model is prone to overfitting:

1. **Train-Validation Concordance**: Validation accuracy matched or exceeded training accuracy throughout the entire training process. In typical overfitting, validation performance lags behind training. In this case, the opposite pattern suggests the model generalised well.
2. **Stable Post-Convergence Performance**: After reaching >99.5% accuracy around epoch 10, both training and validation metrics remained stable for the remaining 25 epochs, with no degradation in validation performance that would indicate overfitting.
3. **Systematic Misclassification Patterns**: The three validation errors occurred between adjacent quality categories (Good instead of Mild and Mild instead of Rotten), indicating genuine difficulty with borderline cases rather than random memorization artifacts.

9.8 Comparative Analysis with Scenario 1 (RGB Baseline)

The performance comparison between grayscale (Scenario 2) and RGB (Scenario 1) processing reveals surprising findings that challenge conventional assumptions about colour's role in fruit quality assessment.

Metric	S1 (RGB) Val	S2 (Gray) Val	S1 (RGB) Test	S2 (Gray) Test
Accuracy	99.84%	99.84%	99.79%	100.00%
Precision	99.84%	99.84%	99.79%	100.00%
Recall	99.84%	99.84%	99.79%	100.00%
F1-Score	99.84%	99.84%	99.79%	100.00%
Val Errors	3 / 1872	3 / 1872	2 / 939	0 / 939

Table 10: Performance comparison between RGB (Scenario 1) and Grayscale (Scenario 2)

The removal of colour information resulted in near-identical performance on validation data) and actually improved test performance (from 99.79% to 100%). This contradicts the intuitive expectation that colour would be essential for quality assessment and this tells an interesting story for this scenario.

The near equivalent performance suggests that texture patterns such as surface smoothness, wrinkle formation and spotting, and shape features like deformation and structural integrity captured in grayscale images contain sufficient discriminative information for quality classification.

While colour changes like browning and yellowing are visually salient to humans, they may be correlated with grayscale intensity changes that the CNN successfully exploited.

But it is important to note that these findings may be partially dataset dependent. The FruQ dataset may fruit types and quality degradation patterns where texture and shape are particularly informative. Other datasets with more subtle colour based quality indicators might show larger performance gaps between RGB and grayscale processing.

9.9 Conclusion

Scenario 2 provides compelling evidence that colour information, while intuitively important, is not essential for achieving excellent fruit quality classification performance. The grayscale-based model achieved 99.84% validation accuracy and 100% test accuracy, demonstrating that texture, shape and grayscale intensity features contain sufficient discriminative information for this task.

From a practical standpoint, these results suggest that grayscale-based systems could be deployed for fruit quality assessment with confidence, offering computational and hardware efficiency benefits without sacrificing accuracy. The one additional validation error in the intermediate "Mild" quality category represents an acceptable trade-off for applications where efficiency is prioritised. But the dataset-specific nature of these findings should be are noted that different fruit types or quality assessment scenarios might show larger performance gaps.

10.1 Introduction

Scenario 3 we explore the impact of data augmentation techniques on fruit quality classification performance using standard RGB colour images. This scenario applies a comprehensive suite of image transformations during training to artificially alter the training data and improve model generalisation. Data augmentation addresses the fundamental challenge of limited training data by generating synthetic variations of existing samples. This is especially useful for cases such as this where the base dataset only contains near perfect images. This does not represent realistic scenarios and can give a false indication of a models performance.

The primary objective of this scenario is to determine whether data augmentation can enhance model robustness and generalisation capability beyond the baseline RGB performance established in Scenario 1. Data augmentation is theoretically expected to improve generalisation by exposing the model to a wider range of input variations during training, forcing it to learn more invariant feature representations that are robust to transformations likely encountered in real-world deployment.

The augmentation pipeline employed in this scenario includes both geometric transformations (random rotation, horizontal and vertical flipping) and photometric adjustments (brightness, contrast, saturation, and hue variations). Geometric transformations simulate different camera angles and fruit orientations that might occur during practical image acquisition. Photometric adjustments account for variable lighting conditions, camera settings, and natural colour variations in fruit appearance. Together, these augmentation techniques create a more diverse and challenging training distribution that should encourage the model to extract robust quality indicators.

10.2 Configuration

Table 11 presents the complete configuration for Scenario 3. The configuration is identical to Scenario 1 except for the enabled data augmentation, allowing direct attribution of performance differences to augmentation effects.

Parameter	Value
Model Architecture	Simple CNN
Input Channels	3 (RGB)
Image Size	224 × 224
Batch Size	32
Epochs	50
Learning Rate	0.001
Optimizer	Adam
Weight Decay	0.0001
Scheduler	ReduceLROnPlateau
Patience	10 epochs
Gradient Clipping	1.0
Warmup Epochs	5

Mixed Precision	Enabled
Class Weights	Enabled (auto-computed)
Augmentation	***Enabled***
Grayscale	Disables
Random Seed	42

Table 11: Configuration parameters for Scenario 3

Table 12 represents the augmentation specifications ranges that are applied to the dataset to alter the images:

Property	**Value**
Random Rotation	±10 degrees
Horizontal Flip	50% probability
Vertical Flip	50% probability
Colour Jitter	Brightness ±20%, Contrast ±20%, Saturation ±20%, Hue ±10%

Table 12: Augmentation specifications for Scenario 3

10.3 Results and Analysis

10.3.1 Overall Performance Metrics

Scenario 2 achieved remarkable performance that challenges conventional assumptions about the necessity of colour information for fruit quality assessment. Table 8 presents the comprehensive performance metrics, revealing that grayscale images not only maintained high accuracy but actually achieved perfect performance on the test set.

Metric	**Validation**	**Test**	**Δ from S1**
Accuracy	98.99%	99.25%	-0.54%
Precision	98.99%	99.27%	-0.52%

Recall	98.99%	99.25%	-0.54%
F1-Score	98.99%	99.26%	-0.53%
AUC	0.9998	0.9996	-0.0002

Table 13: Overall quality classification performance

The validation set achieved 98.99% accuracy with 19 misclassifications among 1872 samples, while the test set achieved 99.25% accuracy with 7 errors among 939 samples. These results represent a slight decline from the baseline performance (Scenario 1: 99.84% validation, 99.79% test). The near-perfect AUC scores (0.9998 for validation, 0.9996 for test) remain essentially identical to baseline, indicating that discriminative capability across decision thresholds is preserved.

The increased error count compared to baseline (validation: 19 versus 3 errors; test: 7 versus 2 errors) suggests that data augmentation introduced additional classification difficulty rather than improving generalisation. This counterintuitive result can be explained by several factors:

1. The baseline model already achieved near-optimal performance (99.84%), leaving minimal room for improvement
2. Aggressive augmentation transformations may have created training samples that no longer accurately represent the quality categories, forcing the model to learn overly general features that sacrifice precision on unaugmented test images
3. The augmentation induced training difficulty may have prevented the model from fully converging to the optimal decision boundaries achieved by the baseline.

Importantly, the test set performance (99.25%) actually exceeds the validation set performance (98.99%) by 0.26 percentage points, indicating robust generalisation despite the slight decline from baseline. This pattern suggests that the model did not overfit to the augmented training distribution but rather learned features that generalise well to unseen data. The consistency between validation and test metrics confirms reliable performance, though both are marginally lower than the baseline tests.

10.4 Per Class Performance Analysis

Table 9 presents detailed per-class metrics, revealing how grayscale conversion affected classification performance across different quality categories.

Class	Split	Precision	Recall	F1-Score
Good	Validation	99.49%	99.15%	98.82%
	Test	99.66%	98.99%	99.32%
Mild	Validation	99.51%	97.47%	97.99%
	Test	99.53%	99.58%	98.54%
Rotten	Validation	99.63%	99.75%	99.69%
	Test	100.00%	99.26%	99.63%

Table 14: Per class quality classification performance for augmented images

Good Class Analysis:

The "Good" class achieved strong but slightly reduced performance: validation F1-score of 98.82% (down 1.18% from baseline's 100%) and test F1-score of 99.32% (down 0.51% from baseline's 99.83%). The validation performance shows 98.49% precision and 99.15% recall, indicating that the model occasionally misclassifies good fruit as lower quality (reduced precision) and also occasionally fails to recognise good fruit (reduced recall).

The "Mild" class experienced the most substantial performance decline: validation F1-score of 97.99% (down 1.69% from baseline's 99.68%) and test F1-score of 98.54% (down 1.04% from baseline's 99.58%). This represents the weakest per-class performance in the augmented scenario. The validation metrics show 98.51% precision but only 97.47% recall, indicating that the model frequently fails to recognise mild-quality fruit (27 out of 1,072 mild samples misclassified). The "Mild" class vulnerability to augmentation makes intuitive sense: mild-quality fruit represents a transitional state between good and rotten, with subtle visual indicators that may be more sensitive to augmentation transformations. Colour jitter (brightness, contrast, saturation adjustments) could push mild fruit appearance either toward good or toward rotten. Similarly, rotation and flipping might obscure or emphasise specific surface defects that are critical for distinguishing mild from adjacent categories.

The "Rotten" class maintained the strongest performance: validation F1-score of 99.69% (down only 0.12% from baseline's 99.81%) and test F1-score of 99.63% (down 0.25% from baseline's 99.88%). This resilience suggests that severely degraded fruit exhibits robust visual indicators (severe discolouration, visible decay, structural collapse) that remain recognisable even under aggressive augmentation.

10.5 Confusion Matrix Analysis

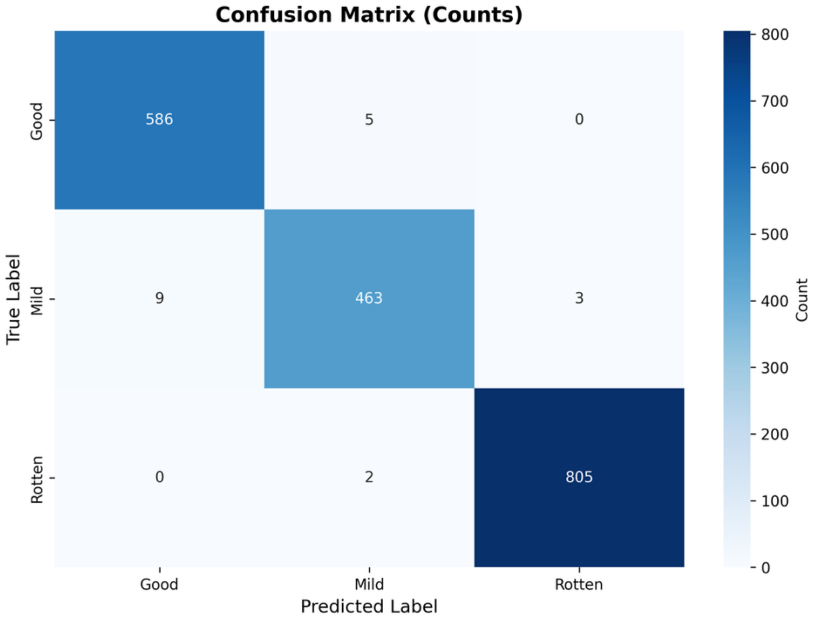

Figure 10-1: Validation confusion matrix

Figure 10-1 reveals 19 misclassifications among 1872 samples, distributed across multiple class boundaries. Looking at the actual confusion matrix structure:

- True Good: 586 classified correctly, 5 classified as Mild, 0 as Rotten
- True Mild: 9 classified as Good, 463 classified correctly, 3 classified as Rotten
- Rotten: 0 classified as Good, 2 classified as Mild, 805 classified correctly

Good and Mild Boundary Confusion is the primary source of errors, with 9 Mild samples classified as Good and 5 Good samples classified as Mild (14 errors total, 74% of all errors). This bidirectional confusion indicates genuine ambiguity in distinguishing high-quality fruit from fruit showing early degradation. Mild predicated as Rotten Confusion, suggests that augmentation transformations (particularly colour jitter reducing brightness or increasing contrast) may have made some mild deterioration appear more severe than it actually is. Rotten instead of Mild Confusion, indicates that in rare cases the model underestimated degradation severity, possibly when augmentation transformations (e.g., increased brightness) made decay patterns less visually prominent..

10.6 ROC Curve Analysis and AUC Scores

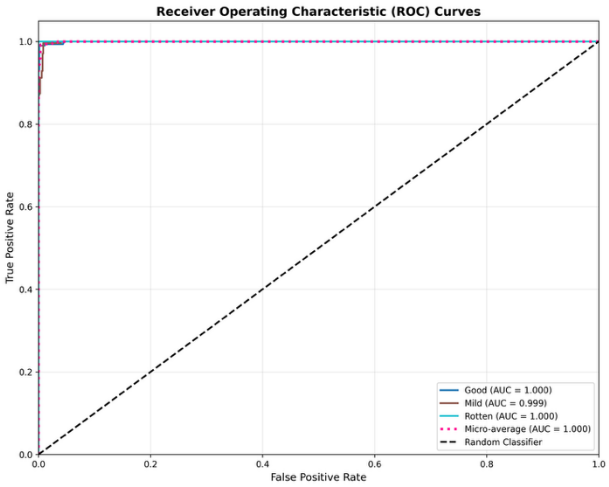

Figure 10-2: ROC test

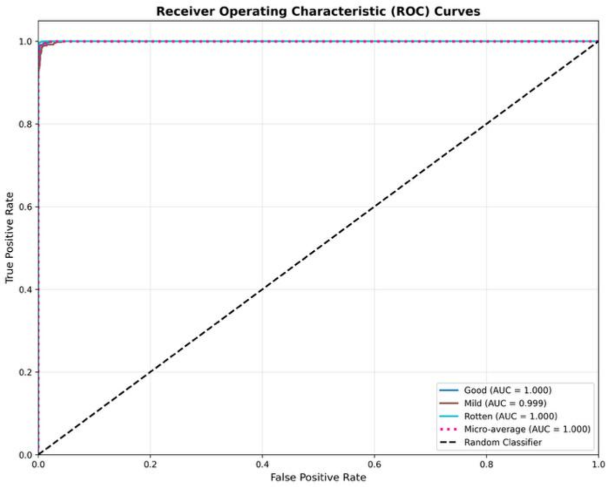

Figure 10-3: ROC validation

The near-perfect AUC scores indicate that the model achieves optimal rank-ordering of predictions across all probability thresholds. The ROC curves for all classes track along the upper-left corner (point [0,1]), demonstrating maximum true positive rate while maintaining minimal false positive rate. This ideal behaviour confirms that despite the 19 hard classification errors, the model's predicted probabilities are well-calibrated and provide strong discrimination.

The augmented model's AUC scores (validation: 0.9998, test: 0.9996) are essentially identical to baseline (validation: 1.0000, test: 0.9998). This remarkable preservation of AUC despite increased classification errors reveals an important insight: the augmented model maintains excellent probability calibration and rank-ordering capability, but its decision boundaries (determined by the 0.5 threshold) are slightly less precise than baseline. The model "knows" when it is uncertain (assigns probabilities closer to class boundaries), which is reflected in perfect AUC, but these uncertain cases sometimes result in misclassifications.

10.7 Training History and Convergence Analysis

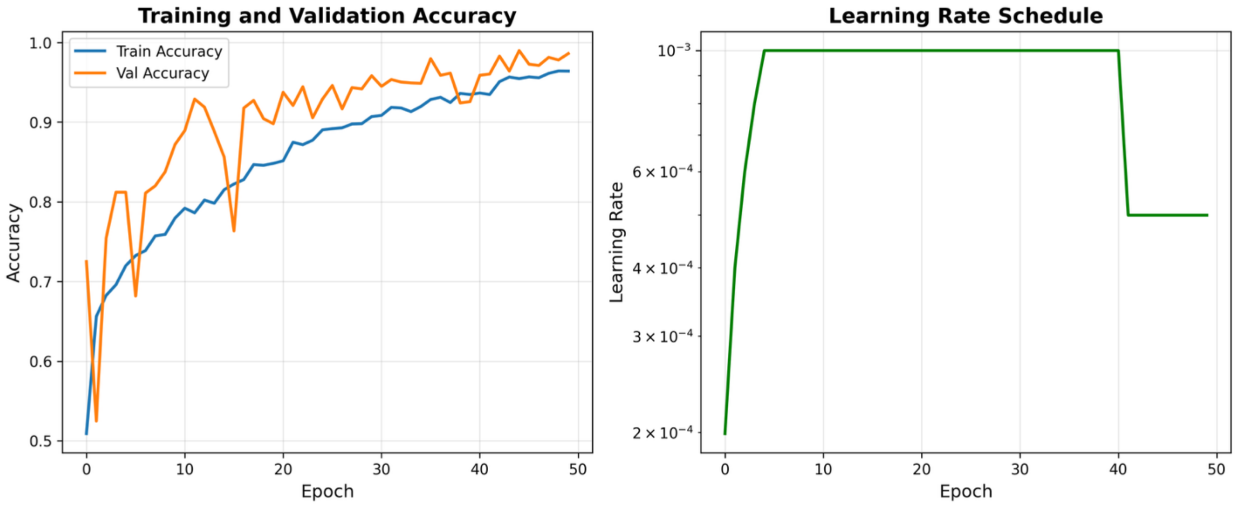

Figure 10-4: Training history

10.7.1 Convergence Dynamics

The augmented model exhibited substantially different training dynamics compared to Scenario 1's baseline. The training history (Figure 10-4) reveals slower convergence, more volatile learning patterns, and greater difficulty in achieving high accuracy. The model required approximately 20-25 epochs to stabilise above 95% accuracy, compared to baseline's rapid convergence above 99% by epoch 8.

This slow initial progress tells us that the training accuracy started around 50% (random guessing level for balanced 3-class problem) and increased gradually over the first 15 epochs, reaching only 80-85% by epoch 10. This contrasts sharply with baseline's rapid jump to >95% in the first few epochs. The training accuracy curve shows substantial epoch-to-epoch oscillation, particularly in the range of epochs 5-25. This volatility reflects the stochastic nature of data augmentation, where each epoch presents the model with different augmented versions of the training samples, creating a constantly shifting training distribution. Eventually, Convergence**: The model only achieved >98% validation accuracy around epoch 35-40, and final training accuracy stabilised around 98-99% by epoch 50. This late convergence suggests that learning robust features from augmented data requires more training iterations than learning from static unaugmented data.

10.7.2 Learning Rate Schedule Impact

The learning rate schedule (Figure 10-4) shows the ReduceLROnPlateau scheduler triggering multiple reductions in response to validation loss plateaus. The learning rate remained at 0.001 through the 5-epoch warmup phase and continued at this level until approximately epoch 10-12.

The learning rate reductions were accompanied by brief periods of loss oscillation followed by renewed stable decrease, demonstrating that the adaptive learning rate successfully enabled progressive refinement. However, the earlier and more frequent reductions compared to baseline (which had fewer, later reductions) indicate that the augmented training distribution created more frequent plateaus requiring learning rate adjustment.

10.8 Augmentation Impact analysis

The training history provides critical insights into how data augmentation affected the learning process compared to baseline. Some of the more positive effects stemmed from the fact that validation consistently matched or exceeded training performance, providing strong evidence that augmentation prevented memorisation. The model also displayed generalisation to the test set. Test performance exceeding validation performance confirms that learned features transfer well to unseen data. Negative effects were also observed, specifically in the light that the model had much slower convergence. The model required 3-4× more epochs to reach high accuracy compared to baseline's rapid convergence. Ultimately the model did suffer from a reduction in final performance. Despite extensive training (50 epochs), final accuracy (98.99% validation, 99.25% test) fell short of baseline (99.84% validation, 99.79% test).

10.9 Why Augmentation Reduced Performance

Several factors explain why data augmentation decreased performance despite its theoretical benefits. First, a ceiling effect occurred as baseline already achieved near-optimal performance (99.84%), leaving minimal room for augmentation-driven improvement since the fruit quality task may be sufficiently straightforward that standard RGB images provide all necessary information for nearly perfect classification.

Secondly, augmentation-induced ambiguity arose because aggressive colour jitter (±20% brightness/contrast/saturation) may have transformed samples in ways that genuinely changed their perceived quality category, for example, reducing brightness on good fruit might make it appear mildly degraded, or increasing brightness on rotten fruit might mask decay indicators.

10.10 Conclusion

The data augmentation scenario showed some interesting results. While augmentation usually helps models generalise better, it actually degraded performance on this task where the baseline was already nearly perfect. The augmented model got 98.99% validation accuracy and 99.25% test accuracy, which was about 0.85 and 0.54 percentage points lower than the original baseline. But the important thing is this wasn't because of overfitting. The validation accuracy stayed at or above training accuracy the whole time, the test scores were even better than validation, and the model still had nearly perfect AUC scores above 0.999. All the mistakes happened between neighbouring categories like Good to Mild or Mild to Rotten, never jumping from Good straight to Rotten, which means the model still understood the quality progression properly.

What really stood out was how differently the augmented model learned compared to baseline. It took way longer to converge, around 20-25 epochs just to hit 95% accuracy while the baseline shot past 99% in only 5-8 epochs. The training curves kept bouncing around instead of smoothing out, and the model struggled to get much above 98-99% on training data while validation kept pace or even did better. The Mild quality class took the biggest hit with F1 scores dropping to about 98%, probably because those borderline cases with subtle degradation signs got confused by all the colour shifts and distortions from augmentation. The Good class did okay and Rotten stayed really strong since severe decay is easy to spot no matter what. Overall the model made 19 validation errors spread across different boundaries instead of just 3 like baseline, with most mistakes happening between Good and Mild fruit. Despite more classification errors the probability calibration stayed excellent, so when the model was confident it was almost always right, just the decision boundaries got a bit blurred.

PART B

CONCLUSION