A Lightweight Colour-Aware Apple Counting Framework Based on YOLOv11

Freddie Pitcher, Hazel Li, George Laurie, Thomas Green, Xiaohe Wu
University of the West of England

August 3, 2026

No.	**Name of group members**	**Contribution to project**	**Contribution to report**	**Signature**
1	Freddie Pitcher	Data Augmentation	Section 1	F.S.P.
2	Hazel Li	Image Preprocessing	Sections 2.2.1, 2.2.3, 3.3	Hazel Li
3	George Laurie	Image Preprocessing	2.2 Image Processing, 2.2.2 Red Apple Detection: V-A Merge	G.R.L.
4	Xiaohe Wu	Model Training and Evaluation on MinneApple	Sections 2.3, 3.1, and Appendix	X.H.W
5	Thomas Green	Model Training and Evaluation on AppleBBCH81	Sections 2.1, 3.2 and Conclusion	T.B.G

Contents

1 Introduction

1.1 Project Scope

The primary aim of this project is to design, implement, and evaluate a machine vision algorithm for the automated counting of apples in orchard environments. Unlike controlled indoor settings, agricultural environments present significant challenges, including varied illumination, complex backgrounds, and much higher degrees of occlusion. To address these, this project proposes a colour-aware hybrid pipeline that integrates conventional image processing with machine learning techniques. The system is trained and evaluated on two distinct datasets – MinneApple (Haeni, Roy, and Isler 2019) and AppleBBCH81 (Kodors et al. 2024) – to assess its cross-dataset generalisation capabilities and robustness against domain shift.

1.2 Literature Review

Fruit counting has evolved from manual feature engineering to advanced deep learning methodologies. The following section critically evaluates relevant approaches and their applicability to this project.

1.2.1 Conventional Image Processing

Early approaches to detection of fruit relied heavily on colour space transformations and morphological operations. Converting RGB images to HSV or LAB colour spaces is a standard technique used to separate fruit from foliage based on the quality of colour. While computationally efficient, static colour thresholding is very sensitive to changes in illumination, a fixed masking threshold also often fails due to this (Tang et al. 2023). Shape-based methods, such as the Circular Hough Transform (CHT) have been employed to detect apples based on their spherical nature. However, CHT is computationally expensive and performs poorly when apple boundaries are incomplete or distorted by occlusion or perspective effects (Albarracin et al. 2019).

1.2.2 Machine Learning Methods

Deep Convolutional Neural Networks (CNNs) have become the standard for agricultural vision. Object detection models, such as the YOLO (You Only Look Once) family (Redmon et al. 2016), offer significant advantages over traditional sliding-window classifiers. YOLO treats detection as a single regression problem, allowing it to leverage the full image context and reduce false positives from background noise.

Although two-stage detectors like Faster R-CNN offer high accuracy, single-stage detectors like YOLO are often preferred for agricultural robotics due to their real-time inference speeds (Yan et al. 2024). The recent YOLOv11 iteration utilised in this project offers improved feature extraction capabilities, making it more effective for detecting small, partially occluded objects compared to earlier versions.

2 Methodology

2.1 Algorithm Design

2.1.1 Overall Algorithm

Figure 1 illustrates the proposed pipeline, which dynamically routes images through colour-specific enhancement branches based on a binary classification. The respective processing methods for these branches are described in Sections 2.2.2 and 2.2.3.

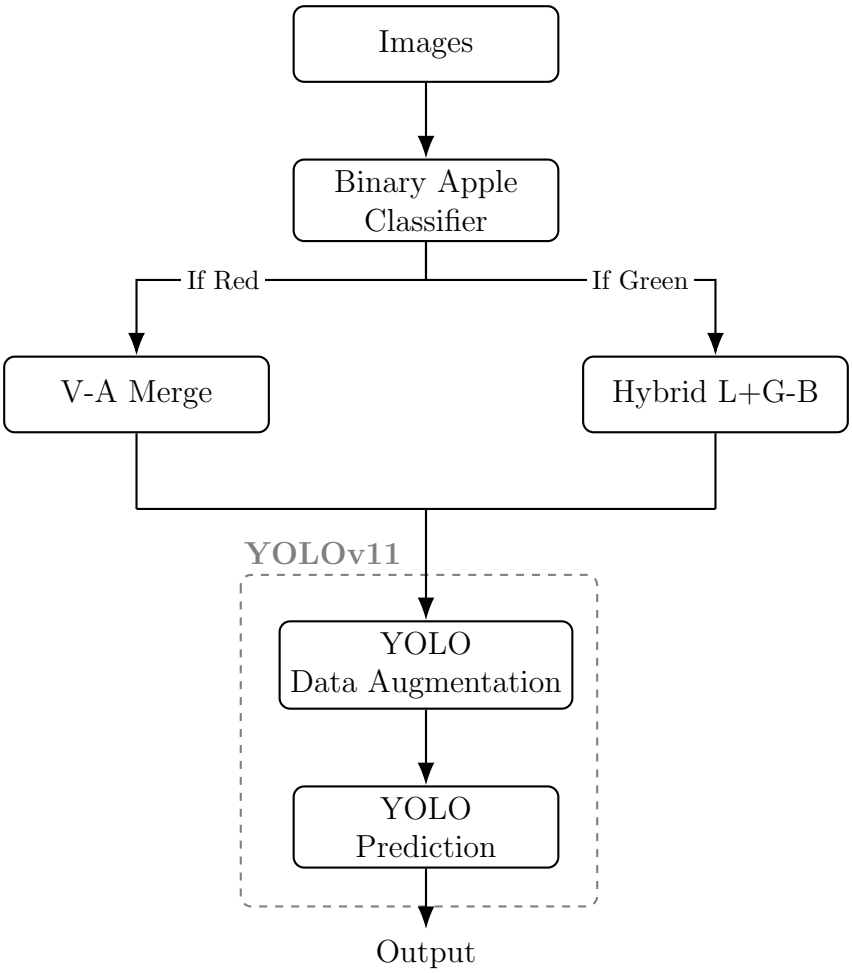

Figure 1: The proposed colour-aware pipeline

2.1.2 Datasets

We used two apple orchard datasets: *MinneApple* (670 images) (Haeni, Roy, and Isler 2019) from the University of Minnesota and *AppleBBCH81* (1,838 images) (Kodors et al. 2024) from the LatHort project.

AppleBBCH81 contains mainly red apples over a narrow maturity range and uses basic bounding box annotations, with many fallen or occluded apples unlabelled. In contrast, MinneApple covers a wider maturity range with both green and red apples and uses high-quality polygon masking, consistently labelling fallen, hanging, and occluded apples. MinneApple is divided into a counting subset and a detection subset, we elected to use

the detection subset because the counting subset lacked ground truth bounding boxes, which limited our ability to identify false positives.

Both datasets were divided into 90% training and 10% testing sets. Owing to its higher data quality, the MinneApple dataset was chosen as the primary training dataset, while the AppleBBCH81 dataset was used mainly to evaluate and improve model generalisation.

2.2 Image Processing

To investigate which image pre-processing techniques best suit this application, while creating a general approach, different colour scales were split from a red and green apple in the MinneApple counting dataset. Figure 2 shows each image split into twelve channels to evaluate their effectiveness in enhancing the apple and suppressing occlusions.

It was determined that for green apples, the value channel in the HSV colour scale produced the most promising result; while the LAB A channel (green-red) most prominently enhanced red apples.

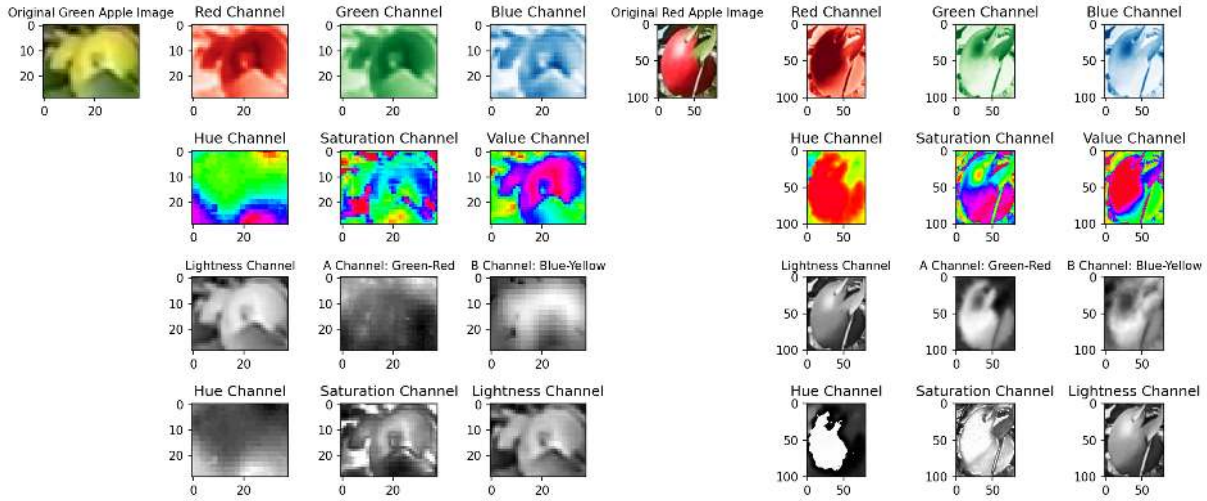

Figure 2: Red and green apple separated into the colour channels Red Green Blue (RGB), Hue Saturation Value (HSV), Lightness Green-Red Blue-Yellow (LAB) and Hue Saturation Lightness (HSL).

2.2.1 Full Pipeline Process: Classifying Red Green

To enable colour-specific image enhancement, we introduced a lightweight binary classifier as the initial stage of the pipeline. This module determines the dominant fruit colour (Red or Green) of the orchard image, allowing the system to dynamically route input frames to the appropriate preprocessing branch.

While simple pixel-ratio thresholds proved unreliable for images containing sparse or small apple clusters, our approach utilises a robust dual-criteria strategy in the HSV colour space. First, a binary red mask is generated by defining "Red" as the wrapped Hue interval ($H < 15 \cup H > 160$) combined with a Saturation threshold ($S > 60$) to suppress shadows and specular highlights. Following morphological opening to remove noise, the algorithm extracts connected components to evaluate two metrics: the area of the largest contiguous red blob (A_{max}) and the global red pixel ratio (r_{red}).

As outlined in Algorithm 2, the classification decision logic is designed to minimise false negatives. An image is classified as **Red** if it satisfies either of the following conditions:

1. **Presence Rule:** A sufficiently large red blob exists ($A_{max} \geq 300$ pixels), ensuring detection of distinct red apples even in low-density scenes.
2. **Density Rule:** The global red pixel ratio exceeds a fallback threshold ($r_{red} > T_{ratio}$), capturing scenarios with many small, scattered fruits.

Failing either condition results in a **Green** classification. This rule-based approach achieved a classification accuracy of 96.41% on the MinneApple dataset, evaluated against ground-truth labels extracted from segmentation masks. Figure 3 provides an example of classification results on both classes of apples.

Figure 3: Examples of red and green apple image classification results.

2.2.2 Red Apple Detection: V-A Merge

As most apples in either dataset are red, it was decided that the emphasis should initially be placed on the $LAB - A$ channel colour scale, as it represents the red-green axis. Red apples have high positive values in the $LAB - A$ channel, creating a strong contrast against the background. The A & B channels are also light-independent, meaning the red-green information remains regardless of shadows or bright spots.

HSV Value channel does not accentuate the red apples as much as $LAB - A$ channel, and struggles under light and dark spots. However, it was included in case the classification algorithm fails, the green apples would still be emphasised against the background.

It was empirically determined that the training data should be a merged image of 80% $LAB - A$ (Green-Red) colour channel, 20% *HSV Value* colour channel, and then original BGR image green and red channels, removing the blue channel.

Figure 4 presents an example application of the red apple detection. Algorithm A.3 in the appendix outlines the process.

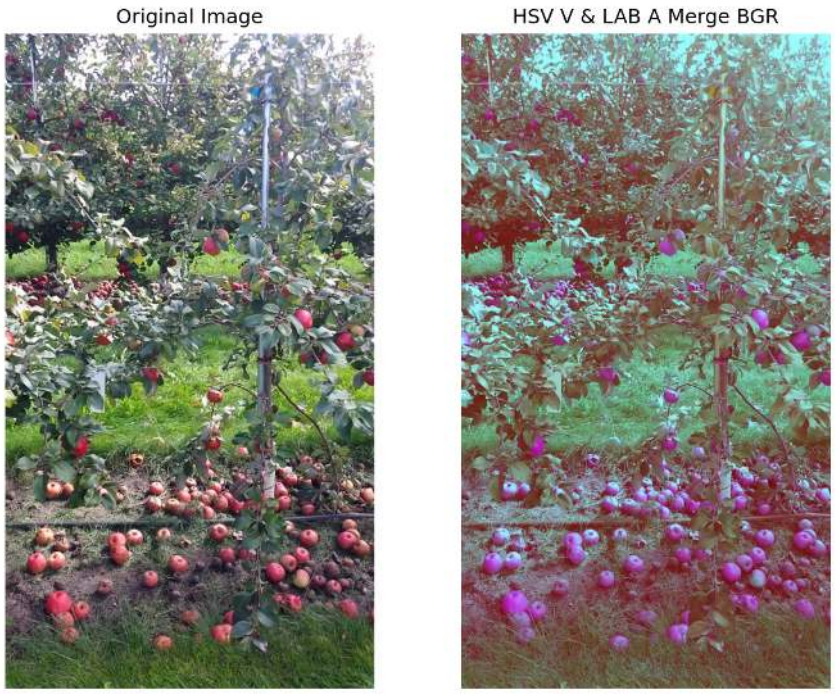

Figure 4: Red apples detection image, merged image of original BGR image, 80% $LAB - A$ and 20% $HSV - V$ combined.

2.2.3 Green Apple Detection: Hybrid L+G-B

Segmenting green apples is inherently more complex than red varieties due to their low spectural contrast to the surrounding foliage. Consequently, the $LAB - A$ channel, while highly effective for isolating red fruit, fails to distinguish green apples from the leaf canopy. A line section analysis was therefore conducted to examine colour differences between apples and background. By sampling RGB values along a line crossing both regions, it was observed that the G channel is consistently stronger than the B channel within green apple regions. This motivates the use of the $(G-B)$ chromatic difference for segmentation (Xia et al. 2017).

To mitigate the impact of variable orchard illumination on green apple segmentation, we employ an adaptive Green-Blue difference metric. Rather than applying fixed coefficients, we dynamically scale the Blue channel based on the ratio of the global channel means $(\bar{x}_B, \bar{x}_G)$:

$$D_{GB} = \text{Norm}\left(G - \frac{\bar{x}_B}{\bar{x}_G} \cdot B\right) \tag{1}$$

By normalising the result, this feature highlights relative green saliency while suppressing intensity variations caused by shadows. However, chromatic difference maps often lose high-frequency edge information. To restore this structural definition, the algorithm incorporates the normalised Luminance channel (L). As detailed in Algorithm 4, the final feature map F is generated via a weighted hybrid fusion:

$$F = \text{Norm}(w_L \cdot L + w_{GB} \cdot D_{GB}) \tag{2}$$

Empirically determined weights of $w_L = 0.3$ and $w_{GB} = 1.6$ are applied to prioritise green chromatic separation while retaining sufficient structural context for the detector (see Figure 5).

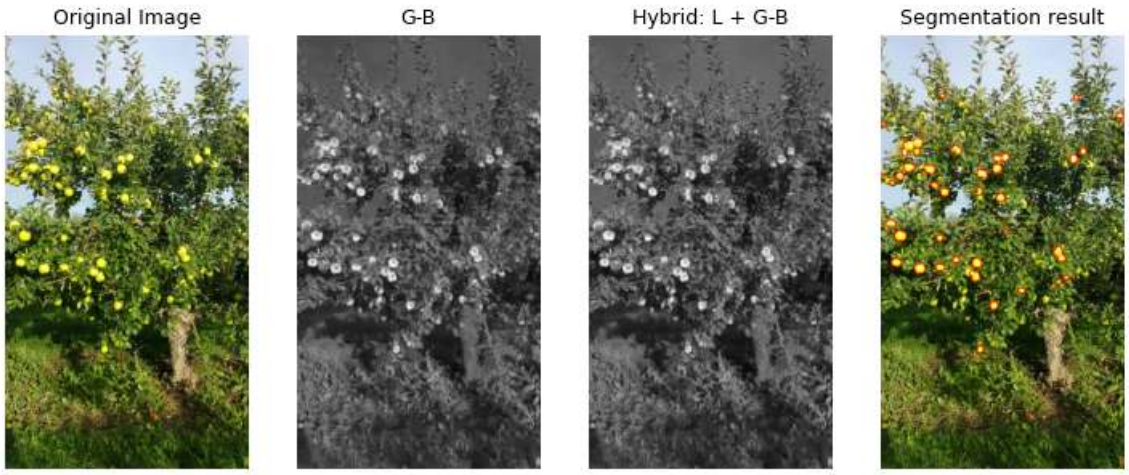

Figure 5: Hybrid L + G–B feature maps for green apple segmentation.

2.3 Machine Learning

2.3.1 Model Selection

Following the colour-aware preprocessing stage described in Section 2.2, the enhanced image frames serve as the input for the object detection framework. YOLOv11 was selected to specifically address foliage occlusion and dense fruit clustering within the MinneApple environment, by utilising the three architectural enhancements relevant to unstructured agricultural environments:

- **C2PSA Attention Module:** Unlike standard convolution, the Cross-Stage Partial Spatial Attention (C2PSA) module applies a spatial re-weighting mechanism. This allows the network to suppress high-frequency textures associated with foliage noise while amplifying features associated with fruit curvature, reducing false negatives in dense canopies.
- **Task-Aligned Learning (TAL):** To mitigate double-counting in fruit clusters, YOLOv11 employs TAL to dynamically align classification scores with intersection-over-union (IoU) accuracy. This strict alignment ensures that the model only converges on anchors that fit the apple geometry perfectly, preventing ambiguous predictions that span multiple small fruits.
- **Multi-Scale Architecture:** The model processes features at different strides (Heads P3–P5), decoupling the detection of foreground apples from background apples. This ensures consistent performance across the significant depth variance found in orchard rows.

2.3.2 Hyperparameter Configuration

To adapt the YOLOv11 architecture to the specific constraints of the MinneApple dataset, several critical hyperparameters were adjusted from their default values. These modifications were driven by empirical ablation studies focusing on small object preservation and occlusion robustness.

Parameter	Value
YOLOv11 Training Configuration	
Input Resolution (`imgsz`)	1280
HSV Augmentation (h, s, v)	0.0
Mosaic Augmentation	1.0
Mixup Probability	0.1
Copy-Paste Probability	0.1
Preprocessing Algorithms	
Apple Classification	
Hue Thresholds (HSV)	$H < 15, H > 160$
Saturation Threshold	> 60
Min Red Area	300 px
Ratio Threshold	0.08
Red Apple Enhancement	
Feature Fusion	$0.2V + 0.8A$
Green Apple Enhancement	
Feature Fusion	$0.3L + 1.6(G - B)$

Table 1: Hyperparameters and configuration settings for the proposed framework.

Standard YOLO training protocols typically default to an input size of 640×640 pixels. However, our dataset consists of high-definition images (720×1280 pixels). Preliminary experiments demonstrated that downscaling to the default resolution resulted in the loss of small fruit instances during feature mapping. Increasing the input size to `imgsz` = 1280 resulted in a 15.38% increase in bounding box Average Precision ($bbox\_AP$) from 0.39 to 0.45 for the baseline model.

We tailored the YOLO data augmentation pipeline (as depicted in figure 1) to complement our preprocessing stage rather than disrupt it:

- **Spectral Preservation (`hsv=0`):** We disabled the standard *Hue*, *Saturation*, and *Value* (*HSV*) jittering (`hsv_h`, `hsv_s`, `hsv_v` = 0). Since our preprocessing algorithm (Section 2.2) relies on precise colour channel arithmetic (e.g., $G - B$ *difference*) to enhance fruit contrast, random colour noise would degrade these engineered features.

- **Small Object Stabilisation (`mosaic=1.0`):** The Mosaic augmentation was maintained at probability 1.0. This technique stitches four training images into a single frame, effectively reducing the relative size of targets and forcing the model to learn robust context features for small objects, addressing the initial training failure where distant apples were frequently missed.

- **Occlusion Robustness (`mixup`, `copy_paste`):** To simulate the heavy foliage occlusion typical of orchards, we introduced Mixup (`mixup` = 0.1) and Copy-Paste (`copy_paste` = 0.1) augmentations. These techniques generate synthetic training samples where apples are partially obscured or blended with leaves, effectively regularising the model against the "*apple − behind − leaf*" scenarios that constitute the majority of false negatives in this domain.

2.3.3 Experimental Model Variants

To evaluate the contribution of each component in the proposed framework, specifically the necessity of the binary classifier and the dual-path preprocessing, we designed an ablation study involving three distinct model configurations:

1. The baseline YOLOv11 model
2. A model utilising only the Red Apple enhancement ($V - A$ *merge*) universally
3. The Full Pipeline which dynamically switches between red and green enhancement strategies based on the binary classifier.

All variants utilise the same YOLOv11 architecture and the optimised hyperparameter set described above to ensure a fair comparison of input feature quality.

3 Results and Evaluation

3.1 Model Performance on MinneApple

Counting accuracy was assessed using Mean Absolute Error (MAE) and Mean Relative Error (MRE). To ensure robust evaluation, we established a fixed confidence threshold of 0.4 for all counting metrics. As illustrated in Figure 6, this threshold corresponds to the global maxima of F1-Confidence curve, effectively mitigating the high frequency of false positive apple identifications observed at lower confidence levels.

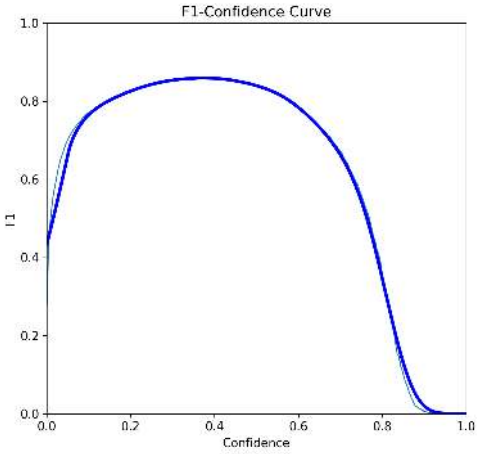

Figure 6: F1-Confidence Curve for the full pipeline

The metrics are defined as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{3}$$

$$\text{MRE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{4}$$

Where N is the total number of test images, y_i is the ground truth count for image i, and $\hat{y}_i$ is the predicted count at conf ≥ 0.4.

As detailed in Table 2, while the Baseline YOLOv11 minimised absolute error (MAE = **6.20**), the Full Pipeline achieved superior relative consistency (MRE = **0.25**). This suggests that the proposed method delivers more proportional counting across variable cluster densities.

Metric	YOLOv11	YOLOv11+V-A merge	YOLOv11+full pipeline
AP [@ 0.5:0.05:0.95]	45.00%	45.46%	**46.03%**
AP [@ 0.5]	**78.82%**	77.11%	78.19%
AP [@ 0.75]	46.25%	47.38%	**48.32%**
AP [small]	30.74%	29.62%	**30.77%**
AP [medium]	60.69%	61.97%	**62.10%**
AP [large]	88.67%	**92.63%**	85.32%
MAE	**6.20**	7.34	7.29
MRE	0.26	0.45	**0.25**

Table 2: Evaluation of model variants on MinneApple.

Regarding detection quality, the Full Pipeline attained the highest overall performance (AP = **46.03%**), outperforming both the baseline (45.00%) and the V-A Merge variant (45.46%). Crucially, this advantage was driven by superior performance on small and medium objects.

Finally, despite the MinneApple test set being dominated by red apple varieties, the Full Pipeline outperformed the red-optimised "V-A Merge" variant in overall metrics. While the V-A Merge variant naturally excelled on large objects (AP[large] = **92.63%**) due to this data bias, the Full Pipeline's overall superiority confirms that the upstream binary classifier successfully identified the red dominance without introducing inference noise. This demonstrates the system's robustness: it maintains high precision on homogenous red data while retaining the architectural capacity for multi-variety detection.

3.2 Zero-shot generalisation performance on AppleBBCH81

Table 3 presents the zero-shot evaluation results of our model variants on the AppleBBCH81 dataset. The performance metrics indicate that the base YOLOv11 model outperformed both the V-A merge method and the full pipeline. This contrasts with the results obtained on the MinneApple dataset, suggesting that the proposed pipeline may be overly specialised to MinneApple and would benefit from improved generalisation.

Metric	YOLOv11	YOLOv11+V-A merge	YOLOv11+full pipeline
AP [@ 0.5:0.05:0.95]	**33.39%**	31.18%	26.46%
AP [@ 0.5]	**72.67%**	66.85%	60.53%
AP [@ 0.75]	**25.86%**	23.48%	18.00%
AP [small]	**0.75%**	0.39%	0.34%
AP [medium]	**37.29%**	36.88%	31.44%
AP [large]	**5.05%**	0.0%	0.0%
MAE	**2.98**	3.51	4.32
MRE	**0.57**	0.68	0.96

Table 3: Zero-shot Evaluation of model variants on AppleBBCH81.

However, it is important to note that the AppleBBCH81 dataset suffers from poor annotation quality. As shown in Figure 7a, only a small proportion of apples are labelled in the ground-truth annotations. In contrast, our pipeline detects substantially more apples in the same image (Figure 7b).

As a result, the MAE and MRE indicate poor performance when evaluated strictly against the provided ground truth. However, this discrepancy, or a portion of it, is likely attributable to annotation incompleteness rather than poor model performance. Therefore, although the quantitative metrics for YOLOv11 with the full pipeline suggest degraded performance, the qualitative results imply that the model may in fact be identifying apples more accurately than the ground-truth labels reflect.

(a) Ground Truth Annotations (b) Predicted Annotations

Figure 7: Comparison of apple detection annotations. (a) Ground truth labels provided by the AppleBBCH81 dataset. (b) Predicted annotations generated by our full YOLOv11 pipeline.

3.3 Reliability Analysis of Binary Classifier

The proposed rule-based classifier demonstrates high robustness, achieving an overall accuracy of 96.41% and an F1-score of 0.96 on the test set. Despite this efficiency, the reliance on hard heuristic thresholds introduces specific edge-case failures, primarily tied to the strictly defined decision boundaries.

False negatives typically arise when the spectral properties of the fruit deviate towards transitional hues (e.g., orange or immature red) that fall outside the strict HSV definition of "Red." As illustrated in Figure 8 (Left), while the algorithm successfully isolates a candidate red region, the largest connected component contains only 286 pixels. Since this falls just below the decision boundary ($Area < 300$) and the global red ratio is negligible (0.0022), the system fails to trigger the "Presence Rule," resulting in an incorrect Green classification despite the visible presence of red fruit.

Conversely, **false positives** are primarily driven by background noise, such as deteriorating leaves or fallen fruit on the orchard floor. Figure 8 (Right) demonstrates a scenario where the classifier is overly sensitive: a small cluster of red background features forms a

connected component of 302 pixels. This marginally exceeds the threshold ($302 > 300$), effectively "tricking" the presence rule and causing the green-dominant image to be processed via the red-enhancement pipeline.

These cases highlight the inherent trade-off in the design: the 300-pixel threshold is aggressive enough to filter out the majority of noise (yielding high aggregate accuracy) but remains vulnerable when valid targets or noise artifacts lie on the immediate margins of the decision boundary.

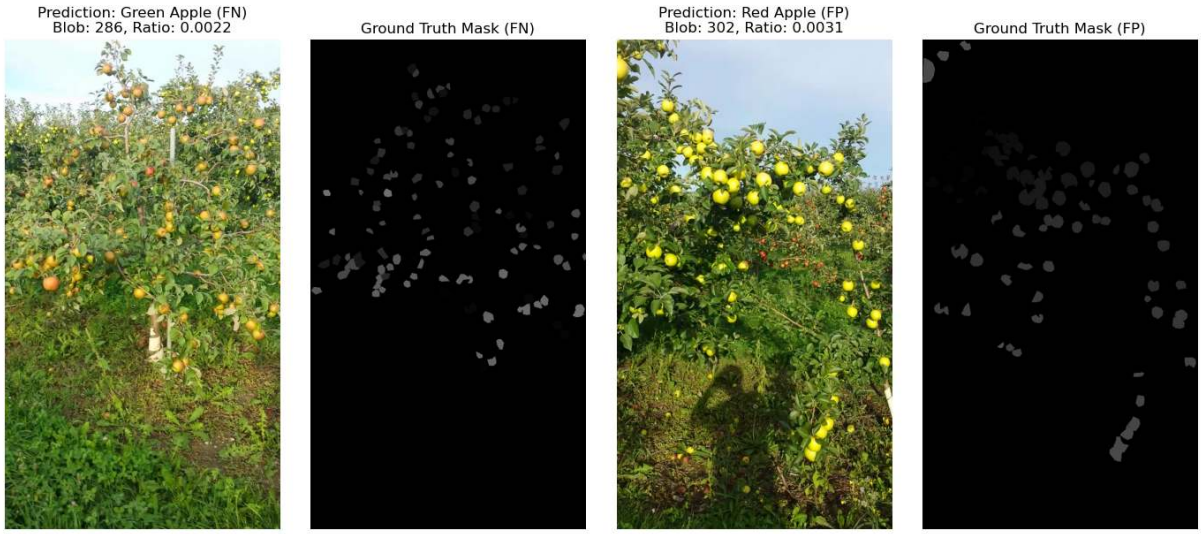

Figure 8: Failure cases of the red–green apple classifier.

4 Conclusion

This study presented a colour-aware hybrid framework for automated apple counting, integrating dynamic image processing with the YOLOv11 architecture. By utilising a binary classifier to route images through spectrally optimised enhancement branches, the proposed pipeline yielded measurable performance improvements on the MinneApple dataset, validating the effectiveness of spectral pre-filtering.

However, zero-shot evaluation on AppleBBCH81 was hindered by poor ground-truth quality, where the model frequently detected unlabelled fruit. Future work must prioritise **data cleaning** and re-annotation to allow for accurate cross-domain benchmarking.

References

Albarracin, J. et al. (2019). "Overlapped Apple Fruit Yield Estimation using Pixel Classification and Hough Transform". In: *International Journal of Advanced Computer Science and Applications* 10.2, pp. 71–76. DOI: 10.14569/IJACSA.2019.0100210.

Haeni, Nicolai, Pravakar Roy, and Volkan Isler (2019). *MinneApple: A Benchmark Dataset for Apple Detection and Segmentation*. Data Repository for the University of Minnesota (DRUM). DOI: 10.13020/8ecp-3r13. URL: https://doi.org/10.13020/8ecp-3r13.

Kodors, S. et al. (2024). "Autonomous Yield Estimation System for Small Commercial Orchards Using UAV and AI". In: *Drones* 8.12, p. 734. DOI: 10.3390/drones8120734. URL: https://doi.org/10.3390/drones8120734.

Redmon, Joseph et al. (2016). "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788. DOI: 10.1109/CVPR.2016.91. URL: https://doi.org/10.1109/CVPR.2016.91.

Tang, Y. et al. (2023). "The Effect of Illumination on HSV Colour Segmentation for Ripe Tomatoes based on Machine Vision". In: *Chemical Engineering Transactions* 114, pp. 829–834. DOI: 10.3303/CET23114139.

Xia, Xue et al. (Aug. 2017). "Detection of Young Green Apples in Orchard Environment Using Adaptive Ratio Chromatic Aberration and HOG-SVM". In: *11th International Conference on Computer and Computing Technologies in Agriculture (CCTA)*. Jilin, China, pp. 253–268. DOI: 10.1007/978-3-030-06137-1_24. URL: https://doi.org/10.1007/978-3-030-06137-1_24.

Yan, B. et al. (2024). "Real-Time Accurate Apple Detection Based on Improved YOLOv8n in Complex Natural Environments". In: *Plants* 14.3, p. 365. DOI: 10.3390/plants13030365. URL: https://doi.org/10.3390/plants13030365.

A Algorithms

A.1 Adaptive Preprocessing Pipeline

Algorithm 1: Adaptive Preprocessing Pipeline

Input: Image I_{BGR}
Output: Preprocessed tensor I_{out}
// Determine apple colour class
$c \leftarrow \text{ClassifyRedGreen}(I_{BGR})$
if $c = Red$ **then**
  $F \leftarrow \text{ProcessRed}(I_{BGR})$
else
  $F \leftarrow \text{ProcessGreen}(I_{BGR})$
end
// Merge feature map with original G, R channels
Extract G, R from I_{BGR}
$I_{out} \leftarrow \text{MergeChannels}(F, G, R)$
return I_{out}

A.2 ClassifyRedGreen

Algorithm 2: ClassifyRedGreen

Input: Image I_{BGR}
Data: Thresholds $T_{hue}^{low}, T_{hue}^{high}, T_{sat}, T_{area}, T_{ratio}$
Output: Class Label $c \in \{\text{Red}, \text{Green}\}$

$I_{HSV} \leftarrow \text{RGB2HSV}(I_{BGR})$
Extract H, S channels from I_{HSV}
`// Identify red pixels (wrapping around 0 and 180)`
$IsRed \leftarrow (H < T_{hue}^{low}) \vee (H > T_{hue}^{high})$
$IsSaturated \leftarrow (S > T_{sat})$
$M_{red} \leftarrow IsRed \wedge IsSaturated$
Refine M_{red} using Morphological Opening
`// Calculate largest red blob and global ratio`
$Blobs \leftarrow \text{FindConnectedComponents}(M_{red})$
$A_{max} \leftarrow \max(\text{Area}(Blobs))$
$r_{red} \leftarrow \text{CountNonZero}(M_{red})/\text{TotalPixels}$
if $A_{max} \geq T_{area}$ ***or*** $r_{red} > T_{ratio}$ **then**
  return *Red*
else
  return *Green*
end

A.3 ProcessRed

Algorithm 3: ProcessRed (V-A Merge)

Input: Image I_{BGR}
Output: Feature Map F

$I_{HSV} \leftarrow \text{RGB2HSV}(I_{BGR})$
Extract V channel
$I_{LAB} \leftarrow \text{RGB2LAB}(I_{BGR})$
Extract A channel
`// Weighted fusion of Value and A-channel`
$F \leftarrow 0.2 \cdot V + 0.8 \cdot A$
return F

A.4 ProcessGreen

Algorithm 4: ProcessGreen (Hybrid L+G-B)

Input: Image I_{BGR}
Output: Feature Map F
Data: Weights $w_L = 0.3, w_{GB} = 1.6$
Extract B, G from I_{BGR}
Calculate channel means μ_B, μ_G
`// Adaptive Green-Blue Difference`
$D_{GB} \leftarrow G - (\frac{\mu_B}{\mu_G}) \cdot B$
$D_{GB} \leftarrow \text{Normalise}(D_{GB})$
`// Luminance for structure`
$L \leftarrow \text{RGB2LAB}(I_{BGR})_L$
$L \leftarrow \text{Normalise}(L)$
`// Hybrid Fusion`
$F \leftarrow \text{Normalise}(w_L \cdot L + w_{GB} \cdot D_{GB})$
return F