Hyperspectral detection, segmentation, and tracking of orchard fruit

1nd Eli Sheppard
Technical Sales
Living Optics LTD
Oxford, UK
Eli@livingoptics.com
0000-0003-4499-8637

2st SeongHyun Cho
Technical Sales
Living Optics LTD
Oxford, United Kingdom
Seonghyun@livingoptics.com
0000-0003-4656-5595

3rd Alex Spanellis
Technical Sales
Living Optics LTD
Oxford, United Kingdom
Alex@livingoptics.com

***Abstract*—This study explores the integration of hyperspectral imaging (HSI) and RGB data for detecting, segmenting, counting, and sizing fruit in agricultural orchards. We present a novel framework combining spatial and spectral information to enhance performance in complex agricultural environments with occlusion, shadowing, and motion blur. Our proposed 1D-ConvNet + SAM2 model demonstrates a 25% improvement in recall over YOLOv8s, highlighting the advantages of incorporating spectral data for detecting partially occluded fruit.**

Using DepthAnythingV2 (DAv2), we estimate metric depth to localize fruit detections within a unified 3D coordinate system, enabling object tracking and unique fruit identification across video frames. Despite challenges such as calibration dependencies and computational overhead, the integration of depth and clustering algorithms has achieved substantial progress in estimating fruit counts.

For fruit sizing, segmentation masks as well as point clouds derived from monocular depth estimation are used to estimate dimensions, albeit with limitations in handling occlusions and segmentation errors. Future work will address these limitations by fine-tuning segmentation and depth estimation models, developing joint spectral-RGB encoders, and exploring better metric depth calibration techniques.

This research emphasizes the potential of spatial-spectral fusion in agricultural HSI applications, offering a public dataset to encourage further advancements in this domain.

***Index Terms*—Hyperspectral image (HSI) classification**

I. Introduction

Hyperspectral imaging (HSI) is an advanced imaging technique that captures a wide range of light wavelengths across multiple narrow bands, offering significant advantages over traditional imaging methods. Unlike conventional imaging, which typically captures images using three primary colours (red, green, and blue), HSI records detailed spectral information across the electromagnetic spectrum, including visible light, infrared, and sometimes even ultraviolet regions.

The core principles of HSI revolve around its high spectral resolution and the concept of spectral signatures. These signatures represent unique patterns that characterise how materials reflect, absorb, or emit light at different wavelengths [43]. By leveraging these detailed spectral signatures, hyperspectral sensors can accurately distinguish between materials, even when they appear visually similar [1]. This capability enables HSI to address a variety of challenges, including material identification [30], [46], chemical composition detection [16], medical diagnostics [21], cultural heritage preservation [33], and monitoring environmental or biological changes [20]—applications that are difficult or impossible with standard imaging techniques.

Despite its advantages, HSI classification faces several challenges. High-dimensional spectral data, limited labelled training samples, and significant spatial variability in spectral signatures complicate accurate classification [8]. Additionally, factors such as atmospheric disturbances, illumination variations, and instrument effects can degrade the quality of HSI data, further hindering classification accuracy [4], [10]. As a result, HSI classification remains a topic of ongoing research.

This paper presents a comparative analysis of traditional machine learning methods and deep learning techniques, specifically focusing on state-of-the-art approaches for image classification. The primary goal is to explore the foundational principles and recent advancements in spectral detection benchmarks within HSI, specifically in the context of fruit data. Whilst we have focused on the specific domain of an orchard, the techniques and framework explored in this work can be applied to any other HSI data domains easily. To that end, our objectives are as follows:

1) **Establish Benchmarks for Spectral Sensing Technology** - define and develop standardised benchmarks to assess the accuracy and effectiveness of spectral sensing technologies used in HSI for fruit data analysis. This will provide a clear framework for evaluating the performance of various HSI systems in tasks such as detecting, classifying, and segmenting fruit.
2) **Highlight the Benefits of Spectral Sensing** - explore and describe the advantages of using spectral sensing technologies in the context of fruit data analysis. This includes improving the precision of fruit detection and classification, enhancing the ability to track fruit growth, and providing valuable insights for agricultural applications.
3) **Identify Efficient Feature Recognition Methods** - investigate more efficient or effective methods for feature recognition by comparing the performance of spectral-based methods with spatial-spectral-based techniques.

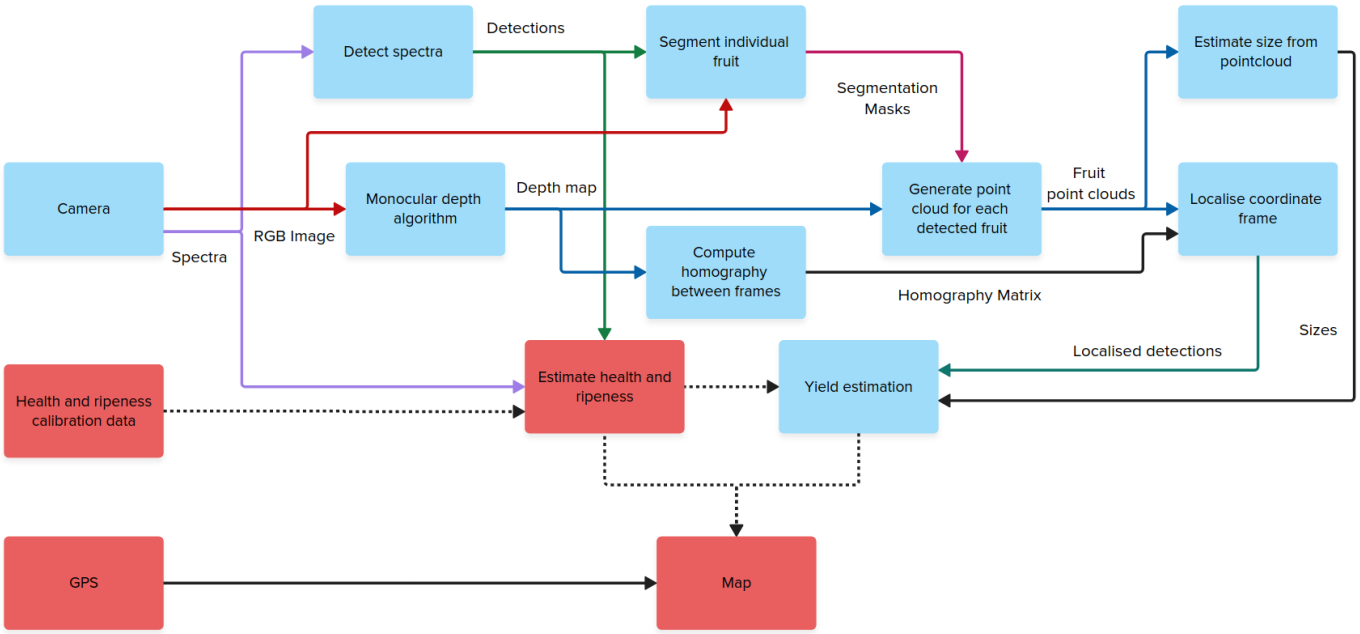


Fig. 1: System overview for a hyperspectral detection, segmentation, tracking and quantification framework. Components in Blue are fully implemented, integrated and tested. Components in Red are beyond the scope of this work and their full development and integration will be the scope of future work.

This will demonstrate the potential advantages of combining spectral and spatial information for better detection and classification of fruit in various environments.

4) **Conduct Empirical Experiments** - perform a series of empirical experiments to compare the performance of algorithms used in fruit data analysis, including tasks such as counting, tracking, and classification. These experiments will be conducted using the same data and evaluation framework to ensure fairness and consistency in evaluating different methods.
5) **Address Real-World Challenges** - identify common problems and propose solutions based on shared results from machine learning approaches applied to fruit data. This will help address challenges such as fruit occlusion, varying lighting conditions, and various environmental factors that can affect the accuracy of fruit detection and classification systems.
6) **Evaluate Performance Metrics** - introduce and evaluate performance metrics that accurately measure the capabilities of HSI systems. Key metrics such as precision, recall, mean Average Precision (mAP), Intersection over Union (IoU), and overall accuracy will be assessed for detecting and classifying specific objects in a real-world setting.

Figure 1 shows an overview of the detection, segmentation and tracking system we implemented for orchard yield estimation which is evaluated in this work.

II. Related Works

A. Evolution of Hyperspectral Imaging Classification Techniques

HSI classification has evolved significantly, starting with traditional machine learning methods such as k-Nearest Neighbors (k-NN) [29], Support Vector Machines (SVM) [32] and Random Forests (RF) [14]. These approaches addressed the challenges of high-dimensional spectral data but were limited by the complexity and volume of the data. Over time, advanced techniques like multinomial logistic regression [25], and deep learning methods—transformers [17], stacked autoencoders (SAEs) [60], and Convolutional Neural Networks (ConvNets) [24]—have gained traction. These newer methods leverage spatial correlations between spectral pixels, improving classification performance when utilised effectively [18].

B. Spectral-Based and spatial-spectral Approaches

HSI classification can be broadly categorised into two approaches: spectral-based and spatial-spectral-based methods. Spectral-based methods focus on the spectral signatures of individual pixels but often neglect spatial dependencies [19]. While they capture detailed spectral information, they may not fully utilise the spatial correlations that are essential for higher accuracy [44]. On the other hand, spatial-spectral methods integrate both spectral and spatial features, leading to enhanced classification results. Among these, ConvNets have shown notable success due to their ability to capture both types of information effectively, especially for complex tasks like fruit classification in agriculture [48], [59].

C. Deep Learning and Recent Advances in HSI Classification

Recent advances in HSI classification have heavily utilised deep learning to achieve superior results. A novel framework was proposed to combine spectral and spatial features using SAEs with Principal Component Analysis (PCA) and logistic regression [8]. This approach demonstrated higher accuracy compared to traditional methods like SVM and PCA classifiers. The study emphasised the importance of hierarchical feature extraction for improved classification, though excessive depth could reduce performance. Deep learning techniques for HSI were reviewed highlighting the challenges of limited labelled data and spectral differences from optical images [2]. The study underscored the effectiveness of 2D and 3D convolutional networks and recommended alternative strategies like unsupervised learning and data augmentation. Li et al. [26] categorised deep learning approaches into spectral-feature, spatial-feature, and spatial-spectral-feature networks, emphasising the need for larger annotated datasets or methods like reinforcement learning to overcome data scarcity.

D. Hyperspectral Imaging in Fruit Quality Assessment

HSI has proven valuable for evaluating fruit maturity and quality attributes, offering a non-destructive method for assessing physical-chemical properties, maturity stages, and decay detection. A review of HSI applications in fruit quality assessment identified challenges in image processing, data mining, and scanning parameters, highlighting the need to integrate spectral and image data for improved practicality [51]. The use of HSI for early decay detection in fruit like apples and citrus has also been explored, with findings suggesting its potential despite challenges such as large data sises, high hardware costs, and redundant data, which limit its industrial implementation [34].

E. Dimensionality Reduction and Quality Assessment Techniques

Research into HSI's ability to assess both external and internal fruit quality has advocated for dimensionality reduction techniques like PCA and Linear Discriminant Analysis (LDA) to simplify data processing, though the high costs of equipment remain a significant barrier [27]. Distinguishing banana maturity stages under varying temperatures has also been demonstrated using HSI, confirming its ability to correlate spectral data with attributes like total soluble solids [39] -during ripening, starch molecules are converted to more soluble sugars such as glucose, fructose, and sucrose. Moreover, studies summarising HSI's capacity to evaluate textural, biochemical, and safety features of fruit and vegetables have underscored its non-destructive nature and potential for real-time detection, while addressing challenges such as morphological calibration and computational demands [37].

F. Innovations in Fruit Ripeness, Maturity and Health Estimation

Studies have expanded the applications of HSI for assessing fruit maturity and ripeness. In one instance, classification of blueberry maturity stages was achieved with band selection techniques like pair-wise class discriminability and hierarchical dimensionality reduction, maintaining high classification accuracy above 88% using classifiers such as k-NN, SVM, and AdaBoost [55]. Citrus canker detection has also been investigated using HSI and spectral information divergence, achieving detection accuracies between 93.3% and 96.7% [62]. Additionally, persimmon ripeness stages have been classified with 95.3% accuracy by applying feature wavelengths and texture features extracted using a linear discriminant analysis classifier [53]. Efforts to automate apple sorting systems have integrated HSI with machine learning techniques for detecting surface lesions, achieving high accuracy in identifying pathogens and showcasing the effectiveness of combining HSI with RGB imaging for enhanced fruit quality control [22].

G. Methods and Challenges in Fruit Counting

Accurate fruit counting is essential for yield estimation, and many traditional and advanced methods have been explored to achieve this goal. For example, a mango yield estimation pipeline employed line-scan HSI on an unmanned vehicle, achieving a determination coefficient of up to 0.83 compared to RGB methods, showing a significant improvement, and demonstrating HSI's multifunctionality, including its potential for disease detection [13]. Automated pineapple crown counting systems have also shown promise, achieving 94.4% accuracy using UAV-captured RGB images and machine learning classifiers like artificial neural networks [45]. Mangos on tree canopies have been successfully counted using texture- and shape-based methods, achieving counts within 16% of actual numbers when imaging conditions were consistent [38].

H. Advances in Machine Learning for Fruit Counting

Research comparing ConvNet architectures for avocado, lemon, and apple detection has shown that Faster R-CNN outperforms single-shot detectors in accuracy but requires more computational resources [49]. Additionally, ConvNets have been used to count clustered apples, achieving 97% accuracy by effectively managing occlusions and varying illumination conditions [15]. A low-cost approach for counting green fruit in orange trees has also been proposed, achieving a detection error of only 5% under controlled conditions [31].

I. Hyperspectral Imaging for Fruit Detection and size Estimation

Recent studies have extended the use of HSI to fruit detection and size estimation. A review of deep learning techniques for these tasks highlighted efforts to address challenges like canopy occlusion and lighting variations [35]. Assessments of internal and external quality attributes in peaches have achieved high accuracy in weight prediction and diameter estimation using HSI, with a minimal margin of error [54]. Combining spectral and spatial data, another study demonstrated accurate fruit size and count estimations, outperforming manual methods [42].

J. Challenges in Yield Estimation Using Hyperspectral Imaging

HSI offers significant promise for yield estimation due to its ability to capture detailed spectral and spatial data. However, several challenges persist that limit its scalability and practical implementation. A key issue is the misclassification of young leaves and overlapping fruit, which results from spectral similarities and canopy occlusion, as noted in studies of early-stage fruit yield estimation [36]. These misclassifications are further exacerbated by environmental variables such as inconsistent lighting, shadows, and field debris, which complicate data acquisition under outdoor conditions.

The computational demands of processing high-dimensional hyperspectral data also present a formidable challenge. Traditional approaches such as linear unmixing have been applied successfully to estimate vegetation abundance and correlate it to yield, achieving reasonable accuracy through multidate imagery analysis. However, these methods struggle with nonlinear effects and require advanced preprocessing to mitigate errors caused by mixed spectral signatures in heterogeneous canopies [28].

Deep learning techniques, such as CNNs, have shown promise for yield estimation by leveraging spectral and spatial information. For example, a CNN-based model integrated spectral and RGB data to predict corn yields with an accuracy of 75.5%, outperforming one-dimensional or two-dimensional models alone. However, the study highlighted the limitations of small sample sises and the dependency on preprocessed data to achieve optimal performance, underscoring the need for larger datasets and robust preprocessing pipelines for model generalisability [58].

In orchard settings, ground-based HSI has been used to estimate mango yields with accuracy comparable to RGB-based methods. However, while HSI offers additional capabilities for detecting nuanced traits like disease and maturity, its cost-effectiveness for standalone yield estimation remains questionable unless integrated into systems addressing multiple agronomic objectives. The pipeline's reliance on labour-intensive ground truthing and extensive image preprocessing also constrains scalability in commercial applications [13].

Furthermore, integrating hyperspectral data with machine vision systems to automate fruit detection across growth stages has shown potential but has limitations; challenges include the need for robust algorithms to handle variability in fruit appearance, such as colour and sise, across different developmental stages and environmental conditions. These systems also face difficulties achieving real-time processing speeds suitable for large-scale agricultural operations [41].

III. Overview

The remainder of this paper is organised into the following sections:

V) **Dataset preparation** - we provide an overview of how the dataset was collected and prepared as well as its contents.

VI) **Fruit detection and segmentation methodology** - an explanation of the methods of detecting fruit we have developed using hyperspectral data and a comparative study between them.

VII) **Yield estimation through object tracking** - an explanation of our detection tracking algorithm and how this can be used to estimate the total number of fruit in a row of trees.

VIII) **Fruit sizing** - a method for estimating the distribution of fruit sises in a row of trees.

IX) **Conclusion** - A discussion of the key contributions of this work

X) **Future Work** - Areas to improve on in future to enhance the utility of HSI workflows.

IV. Dataset Preparation

The dataset was captured using a Living Optics hyperspectral camera [6], which provides detailed spectral data within the visible to near-infrared (VIS-NIR) spectrum, covering wavelengths from 440nm to 900nm. It contains standard RGB images with dimensions of 2048 x 2432 x 3 and hyperspectral data represented by 4384 spectral samples across 96 narrow bands. When combined, we refer to the RGB + spectral data as a hyperspectral (HS) image.

Developed in collaboration with the UK's largest orchard fruit producer, the dataset spans the growing season from March to September and includes over 1,000 trees across multiple fruit varieties. It supports key applications such as fruit counting and sizing, segmentation, and potential disease detection by offering insights into various stages of fruit development.

The labelled portion of the dataset, comprising 44 unique raw files linked to 439 frames, avoids data leakage by employing an 8:2 train-test split at the raw file level, ensuring all frames from a single file are confined to either the training or test set. This method guarantees robust evaluation while maintaining data integrity.

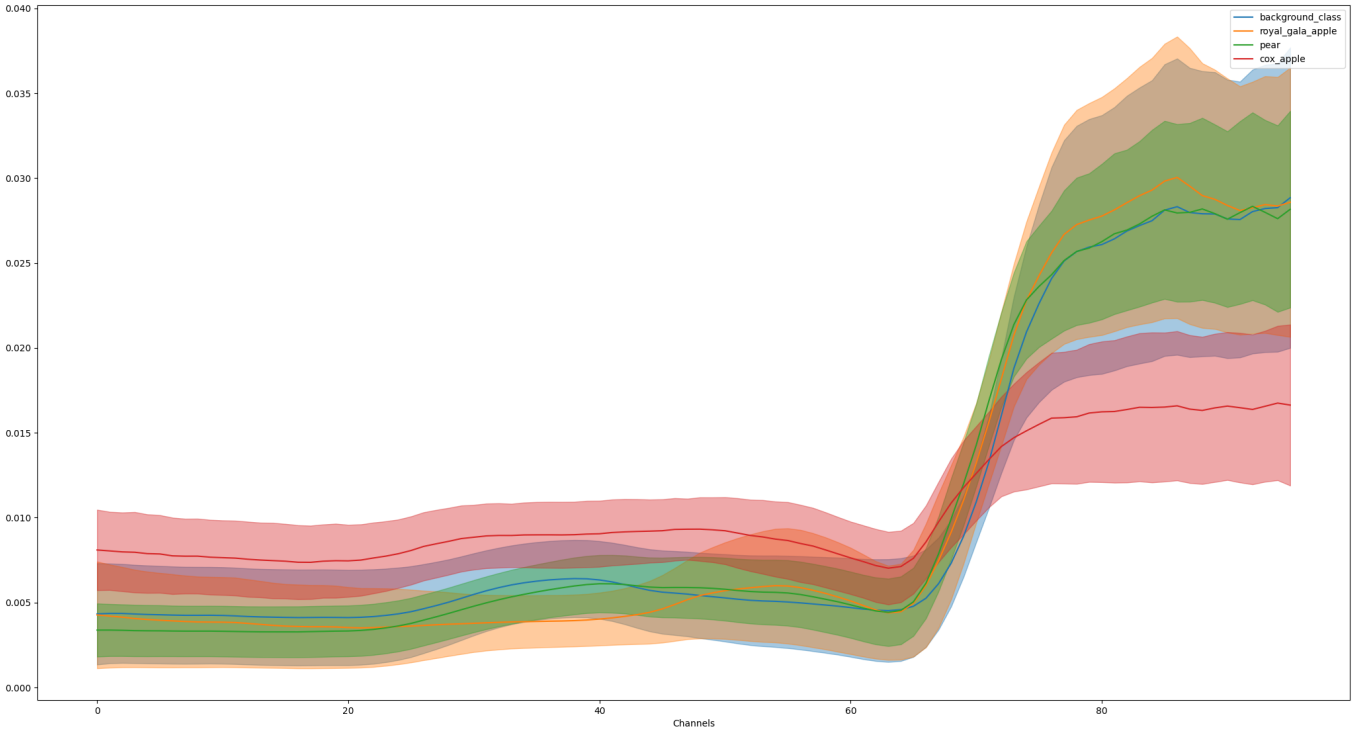
Three distinct fruit classes are annotated: Royal Gala Apple, Pear, and Cox Apple, with their distribution heavily skewed. Specifically, the dataset contains 3,785 instances of Royal Gala Apples, 2,523 instances of Pears, and only 73 instances of Cox Apples, summing to a total of 6,381 labelled instances. This imbalance posed challenges for model training, particularly for the under-represented Cox Apple, requiring careful attention to data balancing techniques during preprocessing.

Due to the lack of Cox Apple instances, these are only present in the training set and not the test set.

We also mine the unlabelled areas of each scene for spectra to form a background class. This further exacerbates the dataset imbalance with a large amount of background data compared to the number of foreground spectra. See section section XI for more details.

Figure 2 shows the mean spectrum and standard deviation of each class in the dataset. It can be seen that the mean pear spectrum is very similar to the background. This is unsurprising as the background is mostly green leaves and

Fig. 2: Mean spectrum of each class in the dataset. The shaded area represents the standard deviation of each class.



the pear variety is also green. The Royal Gala Apples show a distinct bump at around channels 42 to 62, this is what gives them their strong red colouring. The Cox Apples are clearly yellow (and therefore unripe) base don their mean spectrum which shows a broad response across much of the visible spectrum.

A. Labelling spectra

Spectra were extracted from the HS images and each provided with a class label dependant on the class label of the segmentation mask at the location from which the spectral sample was extracted.

This results in the creation of two arrays per data split (train and test). The first array [N, 96] contains an ordered list of spectral samples, the second array [N, 1] contains a list of class numbers. The order of these two arrays is matched, such that the spectrum at (e.g.) index 0 matches the class number at index 0.

For spectra which do not lie within a labelled segmentation mask (i.e. spectra which do not belong to one of the three labelled object classes) a background class label is assign - class number 0.

B. Data normalisation

Two normalisation steps are performed on the spectra in order to maximise the performance of the spectral detection algorithms.

1) Reflectance conversion:
The spectrum of the illuminant of each HS image is estimated by taking the average spectrum of the brightest 5% of spectra within the image. All spectra in the HS image are then divided by this spectrum to convert them from radiance to (pseudo) reflectance. This should reduce the effect of lighting changes between scenes, thus reducing the intra-class variance and making spectral classification easier. See Figure 3 for an example of how this process affects some sample spectra.

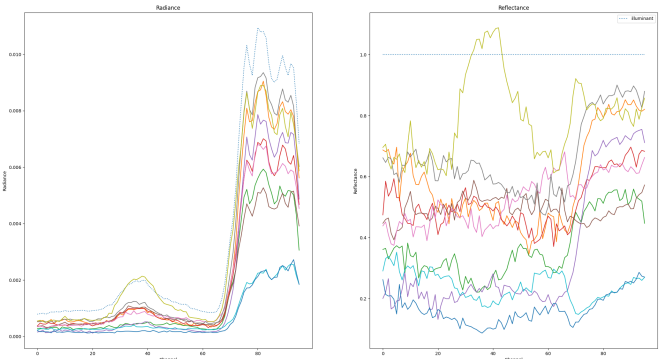


Fig. 3: A selection of spectra and the estimated scene illuminant in radiance (left) and reflectance (right).

2) Sum normalisation:
In order to reduce the effect of intensity variations between HS images and across individual scenes due to changes in camera settings (Gain, Exposure time) as well as shadowing, we divide each spectrum by its sum along the channels axis.

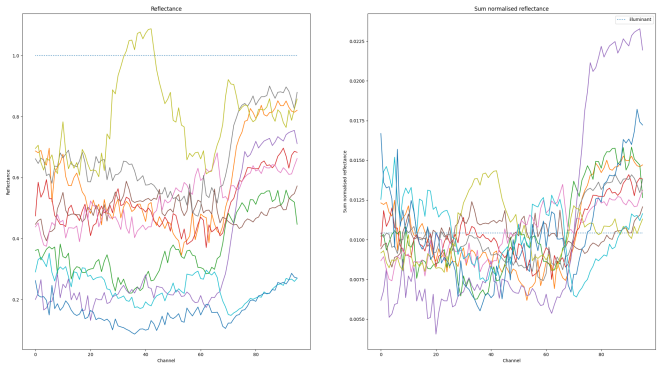


Fig. 4: A selection of spectra and the estimated scene illuminant in reflectance (left) and sum normalised reflectance (right).

This ensures that the area under the curve (AUC) for each spectrum sums to one. This will further reduce intra-class variance. See Figure 4 for an example of how this process affects some sample spectra.

V. Fruit detection and segmentation methodology

A. Experimental Setup

The image size used for these models varies based on the data type, with the Spectral Segmentation Model utilizing the full spectral resolution of 4384 spectral points. The Spatial-Spectral model and YOLOv8s model processes images standardised to a size of 960x960 (cropped and half resolution). This image size is due to a quirk of the Living Optics Camera - spectral samples are only collected for the central (1920 x 1920) region of the image due to the dispersive nature of the optics [6]. As such we only labelled this central region, so any objects visible outside this area in the RGB scene view will not have ground truth segmentation masks.

The experimental setup for this benchmark study focuses on evaluating the performance of seven distinct models in spectral detection tasks using hyperspectral and RGB data. Each model employs a unique strategy, leveraging different data characteristics to optimise detection accuracy and computational efficiency. The models tested include two Spectral detection models, four Spatial-Spectral segmentation models, and an RGB segmentation model, each designed to address specific challenges associated with hyperspectral and RGB image analysis.

B. Spectral detection algorithms

1) Random Forest: The first Spectral detection model utilises a RF algorithm as a baseline to classify hyperspectral data, using only spectral information. We make use of the scikit-learn ensemble and multiclass packages to provide the RF algorithm, and training and inference methods. The RF classifier is trained using the 'OneVsRestClassifier' object from scikit-learn. This allows us to produce a classifier which selects the most likely class for each spectrum it classifies. We used the 'RandomForestClassifier' with custom hyperparameters: *min_samples_leaf*=2, *min_samples_split*=5, and *n_estimators*=400, in addition to exploring the effects of class weighting to address data imbalance, as described in Section **??**.

This leads to an over-fit RF as each tree is extended until the final node is entirely pure (i.e. the training data is 100% correctly classified). However, attempting to limit over-fitting by altering the RF maximum depth leads to worse performance on the test set. A simple explanation of this is that the reflectance conversion and sum normalisation, as described in the subsection IV-B section, have done their job and have lead to the creation of highly separable classes.

2) 1D convolutional classifier: The second Spectral classification model utilises a 1D Convolutional Neural Network (1D-ConvNet) to classify hyperspectral data.

In order to find a suitable architecture and set of hyperparameters for the 1D-ConvNet, a grid search is performed. The basic skeleton of the architecture consists of a single 1D convolution layer applied across the 96 channels of the input spectra followed by a Relu activation function to introduce non-linearity, batch normalization to remove the mean shift introduced by the Relu activation and a 50% dropout layer [52] to mitigate the risk of over-fitting, followed by a single fully connected layer with 4 output neurons followed by a Sigmoid and Softmax activations. During the grid search we add both convolutional and fully connected layers to the skeleton to find the optimal structure for the architecture.

The grid search is performed over 2 parameter sets:

1) the number and size of the convolutional layers for between 1 and 4 convolution layers and sises [64, 128, 256, 512] neurons.
2) the number and size of the fully connected layers between 0 and 4 additional layers (not including the output layer with 4 neurons). Both increasing layer widths and decreasing layer widths are tested (e.g. [64, 128, 256, 512] and [512, 256, 128, 64]

This gives a total of 32 architectures to train and test.

The grid search is run using the 'Background subsampling 1' data balancing method as described below. Batch size is set to 4384, an entire frames worth of spectra, and optimisation is done using Adam [23] with the default parameters. An early stopping patience of 100 epochs is applied, operating on the validation F1 score of the model. The best architecture is the model which achieves the highest validation F1 score. We make the assumption that good any time performance means good end performance [50] as is commonly assumed in architecture searches.

The architecture consists of a single 1D convolution layer that operates across the channels axis of the input spectra, with sixty four neurons. After the convolution we apply a ReLU activation and batch-normalisation. Following the convolutional block, the network contains four fully connected layers with sizes [512, 256, 128, 64] before the output layer. The final output layer contains four neurons, with Sigmoid and

Softmax activation functions applied to produce the desired class probabilities.

One of the major challenges with training the 1D-ConvNet is the data imbalance in the orchard dataset. Imbalance comes in two forms 1) the disparity between the amount of background spectra vs foreground spectra and 2) the disparity between the number of spectra in each of the foreground classes. To address this, we explore a variety of techniques.

1) **None** - the data is used as is with no attempt to account for data imbalance
2) **Class weighting** - the loss is weighted inversely proportional to the frequency of each class such that rare classes are up-weighted and common classes are down-weighted. I.e. getting a rare class wrong is more heavily penalised than getting a common class wrong. Due to the common class being seen more often, its total contribution to the loss should (approximately) equal the contribution of the rare class(es).
3) **Background subsampling 1** - in each epoch a subsample of the background data is selected such that the number of background spectra is equal to the total number of foreground spectra, summed across all foreground classes.
4) **Background subsampling 2** - in each epoch a subsample of the background data is selected such that the number of background spectra is equal to the number of spectra in the rarest foreground class.
5) **Data duplication 1** - The foreground spectra are duplicated to match the number of spectra in the background class.
6) **Data duplication 2** - The spectra in each class are duplicated to match the number of spectra in the background class.

Full details of these methods can be found in section XI.

We retrain the best architecture found by the grid search from scratch using each of the different data balancing methods described above. The results of this experiment can be found in Table I. We use an early stopping patience of 500 epochs to ensure the models have fully converged.

The best data balancing method was 'Background subsampling 1'. ‡Whilst applying no data balancing, achieved 97% accuracy, this is due to the imbalance of the dataset - the classifier simply learned to always predict 'background' for all spectra and failed to learn anything of use. This highlights the importance of both data balancing and the use of metrics beyond accuracy to validate machine learning models.

Further to this, the loss weighting approach also failed to learn, predicting 'background' for all spectra. This suggests that there is an issue with the capacity of the network to accurately model the distribution of background spectra whilst identifying the features of the foreground classes. Potentially increasing the size of the model - which only has a single convolution layer, might make the loss weighting approach more viable, additionally decreasing the weighting of the background class further may also help.

The use of data duplication worked to some extent, but the model still heavily favoured the 'background' class for both data duplication methods. The use of a duplication method in conjunction with spectral augmentation may improve the generalisation of the model. In particular, duplicating the spectra of each foreground class (Data duplication 2) to match the number of background spectra showed promising results. A major drawback of any data duplication strategy is that it increases training time.

We should note that the architecture was optimised using the 'Background subsampling 1' data balancing method. So it is unsurprising the the model performs well with this data balancing method. Ideally, an architecture search across all possible architectures and data balancing methods would be run, however this is prohibitively expensive in terms of computational costs and training time when running a grid search. The use of a more sophisticated search algorithm could facilitate the joint optimisation of more parameters over wider ranges in a more timely manner but we will leave this for future work.

In all following sections, the 1D-ConvNet is the model trained using the Background subsampling 1 data balancing method and architecture found from the grid search.

C. Spatial-spectral segmentation algorithms

The spatial-spectral approach leverages spectral classifiers and spatial segmentors to enhance detection performance for HSI. The methodology evaluates four distinct combinations of spectral and spatial models:

1) RF + FastSAM
2) RF + SAM2
3) 1D-ConvNet + FastSAM
4) 1D-ConvNet + SAM2

The spatial-spectral segmentor is a two stage algorithm which requires first classifying the spectra with a spectral classifier, then over-segmenting the RGB image with a segmentation algorithm. We then combine the locations of the spectral classifications with the segmentation masks to find the segmentation masks of objects of interest.

For spectral classification, both the RF and 1D-ConvNet classifiers learn features directly from the spectral data. RF achieves this through an ensemble of decision trees that collectively identify patterns and relationships within the spectral input. At the same time, the 1D-ConvNet employs a deep learning architecture to extract hierarchical features from raw spectral data. Although their approaches differ—RF using decision tree ensembles and 1D-ConvNet leveraging convolutional and fully-connected layers—both are fundamentally data-driven, learning features based on the patterns in the hyperspectral data.

For spatial segmentation, both FastSAM [61] and SAM2 [40] focus on processing spatial data. FastSAM prioritises computational efficiency and speed, making it suitable for scenarios requiring rapid segmentation, while SAM2 emphasises achieving higher accuracy by employing more computationally expensive spatial segmentation techniques.

TABLE I: Results of different data balancing methods on the optimised 1D convolutional classifier architecture.

Balancing scheme	**Validation Accuracy**				
	Background	**Pear**	**Royal Gala Apple**	**Cox Apple**	**Total**
None	**1.0**	0.0	0.0	N/A	**0.97**‡
Class weighting	**1.0**	0.0	0.0	N/A	**0.97**‡
Background subsampling 1	0.89	**0.79**	**0.60**	N/A	0.88
Background subsampling 2	0.16	0.59	0.54	N/A	0.18
Data duplication 1	0.99	0.20	0.06	N/A	**0.97**‡
Data duplication 2	0.96	0.57	0.23	N/A	0.94

The class of a segment predicted by either SAM2 or FastSAM is determined as the most frequent (median) non-background class as predicted by the set of N spectra within the segment. The classification of each spectrum $n \in N$ is the class with the highest classification probability as predicted by the spectral detector (either RF or 1D-ConvNet).

1) RGB detection and segmentation with a fine-tuned YOLO model: The RGB Model is a fine-tuned YOLOv8s model, specifically trained for segmentation tasks using the pre-trained weights provided by Ultralytics. We then fine tuned the model on the orchard dataset. We are then able to run the model to predict bounding boxes.

The dataset was exported to a YOLOv5 segmentation format by extracting the RGB scene view from each of the HS images and converting the segmentation masks to a set of corner coordinates of polygons which outline each of the labelled objects in the dataset.

The RGB scene view and segmentation masks were cropped to the central 1920x1920 region before exporting the data. This was done for two reasons: 1) The labelled data only contains segmentation masks for this region and 2) the Living Optics Camera only provides spectral samples within this region, so cropping makes comparing the different detection and segmentation models easier and fairer.

Fine-tuning was performed using the Ultralytics fine-tuning framework with the following hyperparameters:

1) Maximum training epochs: 10000
2) Early stopping patience: 200 epochs
3) Image dimensions: 960x960
4) Optimiser: Adam (with default parameters)
5) Dropout rate: 40
6) Default data augmentation parameters were used as defined in the Ultralytics framework when calling 'YOLO.train()'

VI. Performance Comparison

This study utilises a comprehensive performance evaluation framework to rigorously assess the capabilities of the various models for both HSI and RGB detection and segmentation tasks. By focusing on spectral and spatial features, the framework highlights the strengths and limitations of each model, offering a nuanced understanding of their detection performance. The metrics employed ensure a robust and multidimensional analysis of the models' performance.

To begin with, Accuracy is used as a fundamental measure, indicating the overall correctness of predictions. While it provides a general overview, more granular metrics such as Precision and Recall are critical, particularly for datasets with imbalanced class distributions. Precision assesses the model's ability to correctly identify positive instances while minimising False-Positves, whereas Recall (equivalent to True Positive Rate) measures its capability to detect all actual positives. Together, these metrics offer a clearer view of detection performance.

The F1 score is calculated to balance Precision and Recall, whereas the F2 score places greater emphasis on Recall, which is particularly important in scenarios where reducing false negatives is prioritised. This distinction is crucial in spectral detection tasks where undetected instances can lead to significant performance gaps.

For localisation performance, the IoU metric is used in two forms: Spectral IoU and Spatial IoU. Spectral IoU evaluates how effectively the model detects spectral features. The Living Optics Camera has a sparse spectral sampling mask consisting of 4,384 points across the scene, thus spectral IoU computes the area of this mask which has been predicted to be a certain class vs the area of the ground truth mask which belongs to that class. Spatial IoU, derived from predicted and ground truth spatial masks (i.e. from the segmentation of the dense RGB view from the Living Optics Camera as opposed to the sparse spectral view), measures geometric accuracy in localisation, providing a broader understanding of spatial detection capabilities. To complement this, mAP is used to summarise precision across multiple thresholds, offering a robust measure of overall detection quality.

Additional metrics include G-Mean [11] and Matthews Correlation Coefficient (MCC) [9]. G-Mean reflects the balance between sensitivity and specificity, essential for achieving equitable performance across classes. MCC is a comprehensive measure that accounts for all elements of the confusion matrix, providing a robust metric for imbalanced datasets.

False-Positve Rate (FPR) and False Negative Rate (FNR) offer valuable insights into error patterns, their complementarity to Recall ensures a deeper understanding of misclassification trends.

Although this study does not focus on real-time applications, we report the estimated GFLOPs of the different algorithms. This metric is indicative of the time required for each model to generate predictions on a given inference platform which remains a relevant factor in resource allocation and practical model deployment.

$$GFLOPs_{RF} \approx \frac{\sum_{n=0}^{n=N} Depth_n \times 96 \times 4384}{10^9} \quad (1)$$

For the random forest we use the calculation found in Equation 1 to estimate the GFLOPs per frame, where 96 is the number of features in a spectrum and 4384 is the number of spectra in a frame. As the Gini impurity splits are already calculated at inference, the computational cost of the random forest is dominated by the number of trees and their depths.

FLOPs for FastSam, YOLOv8, SAM2 and the 1D-ConvNet are calculated using the flop counter from PyTorch.

A. Experimental Results

1) Spectral Results: From the spectral performance results in Table II, the 1D-ConvNet significantly outperformed RF across multiple metrics. It achieved markedly higher Accuracy, Recall, and Spectral IoU, showcasing its robustness in identifying true positives and capturing hyperspectral features. The RF exhibited higher Precision and a lower FPR, reflecting its conservative predictions with fewer false positives. However, its high FNR and low F2 score indicate challenges in detecting true positives, a critical limitation for this application where we wish to get an accurate estimate of the number of fruit which will be harvested.

Comparing Figure 5a and Figure 5b we can see that the RF has a very high FNR, failing to detect many pear spectra, whilst the 1D-ConvNet has a much higher FPR, incorrectly detecting many background spectra and misassigning many pear spectra as belonging to the Cox Apple class.

Individual FP spectral detections are not as impactful in this workflow as FN detections. As the spectral sampling is sparse and we will be performing a spatial segmentation on the RGB image, FP spectral detections are unlikely to propagate to downstream tasks. I.e. FP spectral detections are unlikely to occur in sufficient density to produce FP spaital-spectral detections.

The classification of many Pear spectra as Cox Apple by the 1D-ConvNet indicates that removal of the very limited amount of Cox Apple spectra from the training data may improve performance as the Cox Apple spectra have clearly acted as a confuser for the Pear class.

One of the key challenges for training the spectral detector was class imbalance, especially between foreground and background regions. Whilst custom balancing algorithms were employed to address this for the 1D-ConvNet, incorporating techniques such as background subsampling and foreground duplication, the RF training loop did not support this workflow. The RF was forced to rely on a loss weighting scheme alone, while this helped mitigate some of the bias towards the background class, it could not overcome the data imbalance entirely. The loss weighting scheme was also applied to the 1D-ConvNet, where it also failed to achieve good results. The RF did outperform the 1D-ConvNet for that particular setup as it had a TPR that was non-zero - the 1D-ConvNet suffered mode collapse due to data imbalance, learning to only predict 'background' for all spectra.

The high FNR of the RF is particularly bad when considered in the context of its combination with an RGB segmentation algorithm. Given that a confidence threshold on the spectral classification of a segment is needed to determine the classification of the segment, having a high FNR for the spectral classification, will lead to an even higher FNR rate for the combined system. I.e. a single TP spectral classification is unlikely to be enough to classify a segment as being a TP, therefore the FNR is increased when the spectral and spatial components are combined.

2) Spatial-Spectral Results: Evaluating the four spatial-spectral segmentors and the spatial-only YOLOv8s model revealed distinct strengths and limitations across different approaches. Table III highlights these results, and Table IV focuses on object-specific counting rates.

Among all model configurations, 1D-ConvNet + SAM2 achieved the best results across most metrics. YOLOv8s achieved a strong Spatial IoU and the highest Precision, underscoring its spatial segmentation strengths. However, its lower performance in other metrics, such as Recall and F1 score, limit its use in downstream tasks. The high FNR of YOLOv8s means that any yield estimate in terms of both counting and sizing will be of negligble utility. **More than half of all fruit were missed by YOLOv8s.**

The object-specific detection results in Table IV further demonstrate the advantages of 1D-ConvNet + SAM2, with counting rates of 0.4597 for Royal Gala Apple and 0.7926 for Pear, surpassing all other configurations, including RF-based models and YOLOv8s. These findings affirm that 1D-ConvNet + SAM2 excels in leveraging both spectral and spatial data for precise instance counting. The superior counting performance of 1D-ConvNet + SAM2 can be linked to its higher Recall and F2 score, highlighting its effectiveness in detecting TPs. The RF's higher Precision comes at the cost of significantly lower recall, leading to missed detections and reduced instance-counting accuracy. These results emphasise the importance of recall-focused metrics, such as the F2 score, in evaluating models for tasks requiring accurate TP detection.

The baseline model, RF + SAM2, provided a crucial benchmark for spatial-spectral detection. However, the low TPR of the RF means that downstream tasks will suffer due to many missed detections.

In contrast, the 1D-ConvNet + SAM2 combination performed better by directly learning spectral features from the data. The use of categorical cross-entropy loss and batch normalisation during training provided stability and a data subsampling scheme to improve dataset balance, enabled the 1D-ConvNet to surpass the baseline model across all significant metrics. Additionally, by integrating SAM2 for spatial segmentation, the combination effectively leveraged both spectral and spatial features, offering a more robust and consistent framework for spatial-spectral detection. This demonstrated clear advantages over the RF + SAM2 baseline, particularly in classifying spectral data.

The 1D-ConvNet + SAM2 combination emerged as the best spatial-spectral detector in this study, achieving the highest

TABLE II: Experimental results of spectral classifiers

Metrics	RF	1D-ConvNet
Accuracy	0.5229	**0.9082**
Precision	**0.3059**	0.2622
Recall	0.1569	**0.7840**
F1 score	0.1836	**0.3653**
F2 score	0.1637	**0.5077**
Spectral_IoU	0.1170	**0.2319**
mAP@50	0.2266	**0.3768**
G-Mean	0.2694	**0.8387**
MCC	0.1936	**0.3688**
TNR	**0.9949**	0.5077
FNR	0.8431	**0.2160**
FPR	**0.0051**	0.0895
GFLOPs	1.05	**0.09**

TABLE III: Experimental results of spatial-spectral classifiers

Metrics	RF + FastSAM	RF + SAM2	1D-ConvNet + FastSAM	1D-ConvNet + SAM2	YOLOv8s
Accuracy	0.5585	0.5446	0.5391	0.9253	**0.9372**
Precision	0.4442	0.3920	0.2931	0.5785	**0.7532**
Recall	0.2893	0.2813	0.4388	**0.7080**	0.4586
F1 score	0.3171	0.2978	0.2211	**0.5574**	0.5561
F2 score	0.2947	0.2820	0.2442	**0.5912**	0.4918
Spatial_IoU	0.2531	0.2315	0.1442	**0.4122**	0.4098
mAP@50	0.4572	0.4238	0.5600	**0.7939**	0.7535
G-Mean	0.3722	0.3622	0.3904	**0.7910**	0.6473
MCC	0.3321	0.3086	0.2322	0.5718	**0.5721**
TNR	**0.9977**	0.9971	0.8090	0.9283	0.9970
FNR	0.7107	0.7187	0.5612	**0.2920**	0.5414
FPR	**0.0023**	0.0029	0.1910	0.0717	0.0030
GFLOPS	7.33	5550.52	6.48	5549.56	**6.28**

TABLE IV: Counting rate

Labels	RF + FastSAM	RF + SAM2	1D-ConvNet + FastSAM	1D-ConvNet + SAM2	YOLOv8s
Royal Gala Apple	0.0879	0.0618	0.1545	**0.4597**	0.2541
Pear	0.4822	0.3933	0.5349	**0.7926**	0.4802

performance across key metrics. It recorded a Spatial IoU of 0.4122 and an mAP@50 of 0.7939, showcasing its ability to integrate spectral and spatial features effectively.

While YOLOv8s achieved strong performance in spatial segmentation with a Spatial IoU of 0.4098 and Precision of 0.7532, its low Recall and high FNR make it unsuitable for real-world applications with high levels of occlusion and shadowing.

In contrast, 1D-ConvNet + SAM2 successfully balanced spectral and spatial detection by leveraging deep learning for spectral feature extraction and SAM2's advanced spatial segmentation. The high Recall of this combined model is a good starting point for future detection workflows. However, a significant drawback of this model is the vast computational cost of SAM2.

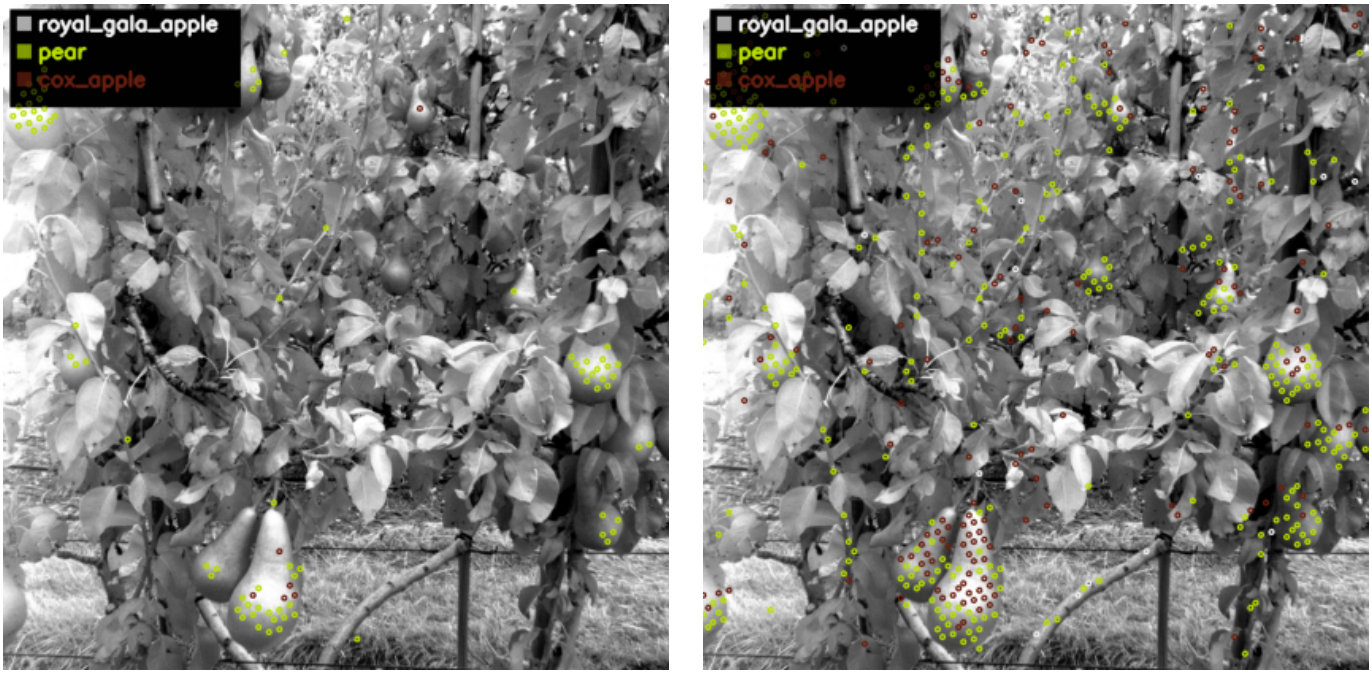
The relatively poor performance of 1D-ConvNet + FastSAM highlights the need to fine-tune the RGB segmentor as YOLOv8s significantly out performed it despite having the same core RGB segmentation model. The cause of the difference is the fine-tuning of YOLOv8s. In future work, we will explore fine-tuning FastSAM to attempt to achieve a performance similar to that of the 1D-ConvNet + SAM2 model with a fraction of the computational resources. Given that YOLOv8s performs better than 1D-ConvNet + SAM2 on some metrics, fine-tuning FastSAM is likely to reveal the true advantages of combing RGB and Spectral data. I.e. vastly improved detection rates with only a very modest increase in computational resource requirements.

In Figure 6a we can see that FastSAM has proposed many incorrect segmentation masks unlike YOLOv8s in Figure 6e which has correctly identified nearly all pears in the scene. This difference is likely the results of the fine-tuing of the YOLOv8s model but not the FastSAM model.

VII. Yield estimation via object tracking

With the development of the spatial-spectral detector complete, we move on to developing a tracking algorithm on top of the best spatial-spectral detector.

Whilst it is trivial to count the number of detections per frame, counting the unique detections across a sequence of frames (a video) is more difficult. We are not guaranteed to detect every object of interest on every frame and movement in the scene, and of the camera relative to the scene, make identifying which objects have already been detected previously difficult.



(a) RF (b) 1D-ConvNet

Fig. 5: Spectral points of detected masks.

We utilise a tracking algorithm to estimate which detections are unique and which are repeat detections of the same object(s).

A. Method

Each detected fruit will be assigned a set of 3D coordinates relative to the position of the camera in the first frame. To do this, we need to estimate the distance between the camera and each detection. I.e. we need a method for estimating a metric depth image.

To do this, we make use of DepthAnythingV2 (DAv2) [56], [57], using the 'vitl' encoder and 'VKitti 2' metric depth checkpoint [5], [12]. These weights are not ideal for the given domain (an orchard) as VKitti 2 contains virtual urban scenes. However, fine tuning DAv2 for the specific domain is beyond the scope of this work.

We set the maximum output depth to 1 as this is just a scaler and will calibrate the depth information in a later step. This means that detections will initially be handled in 'uncalibrated units' i.e. the unacalibrated units are proportional to meters but an additional calibration step is needed to convert between the two measurement systems.

After running DAv2 on the RGB image from the Living Optics Camera, we convert the depth image to a point-cloud in uncalibrated units using Equation 2 on each (X, Y) point within the image.

The segmentation mask of each detected fruit is used to find the coordinates of the centroid of the object i.e. we take the mean X_{ucu}, Y_{ucu} coordinate of the detection and the Z_{ucu} coordinate at this position. Where X_{ucu} is the X coordinate, Y_{ucu} is the Y coordinate and Z_{ucu} is the Z coordinate of the centroid of the segmentation mask in uncalibrated units.

$$P_{ucua} = Z_{ucu}(\frac{P_{pa} - c_a}{f_a}) \tag{2}$$

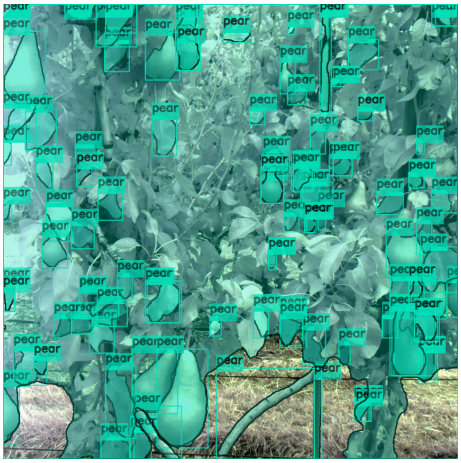
To convert from pixel units to uncalibrated units we use Equation 2 along with the estimated intrinsic parameters of the Living Optics Camera. Where P_{ucu} is the coordinate along axis a (X or Y) in uncalibrated units, $P_{pa} \in [x_p, y_p]$ and f_a is the focal length in pixels for axis a and c_a is the principle point along axis a. The principal point for each axis is assumed to be the central coordinate of the image i.e. $(2048, 2432)/2 = (1024, 1216)$.

$$f_a = \frac{f_{ma}}{PPx} \tag{3}$$

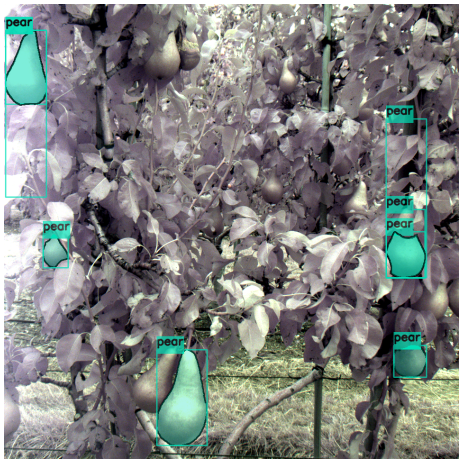
Equation 3 demonstrates the conversion from the focal length of each axis in metres f_{ma} to pixels using the pixel pitch (sise of a pixel), PPx in metres. $PPx = 2.76e^{-6}m$ for the Living Optics Camera and the focal length in the orchard dataset is 8.5mm for both the X and Y axis.

In order to convert from uncalibrated units to meters, we make use of a reference object of known size which is manually labelled in two scenes. The calibration object used were the black squares of a printed chequerboard pattern as shown in Figure 7.

Each black square has a side length 2.5cm, thus we can measure the size of each square in uncalibrated units by extracting their point-clouds using their labelled segmentation masks and use this to derive a calibration factor for converting from uncalibrated units to meters.



(a) RF + FastSAM



(b) RF + SAM2

(c) 1D-ConvNet + FastSAM

(d) 1D-ConvNet + SAM2

(e) YOLOv8s

Fig. 6: Bounding boxes of detected masks.

Fig. 7: The average size of the black squares is used to estimate the conversion from uncalibrated units to meters.

$$L_m = L_{ucu} * J \tag{4}$$

Equation 4 shows how to convert a length L in uncalibrated units to meters using the conversion factor J. $J = 0.025/Mean(L_{square-ucu})$. In practice, we found that calibrating the X and Y axes separately gave better results. This is likely due to the depth image (and RGB image it was derived from) being rectangular, thus we have higher resolution along the X axis than the Y axis. We use $J_x = 15.524$ and $J_y = 15.863$).

There are several limitations to this approach to producing and calibrating a point-cloud with this method.

1) DAv2 is computationally heavy.
2) DAv2 gives different uncalibrated depth measurements depending on the resolution of the input image.
3) The need for a calibration object to be present in the scene reduces the generalisability of this approach.
4) There is no guarantee that the calibration factor, J, is valid for scenes where it was not explicitly calculated.
5) There is no guarantee that the calibration factor, J, is valid at depths other than the depth of the calibration object - DAv2 may produce non-linear depth values.
6) We have not implemented an automatic way of detecting the calibration object, thus this process can only be run offline, with human intervention.

In [57] they make use of a single value for J, which is dependant on the maximum depth of the dataset being evaluated. However, this approach does not generalise to real-world data as shown by our need to derive our own, axis-dependant, calibration factors. However, this does imply that DAv2 should provide linearly varying depth values, thus points 4 and 5 are of less concern than the other issues.

In future, work we will explore how to improve the performance of the monocular depth algorithm in terms of inference speed, depth accuracy, calibration accuracy and calibration method.

In order to unify the coordinate system of all detections, we use Iterative Closest Point (ICP) [3] with a 'Point-ToPlane' transformation [7] to compute the transformation between $frame_n$ and $frame_{n-1}$ which can be used to transform the coordinates of each detection in $frame_n$ back to the coordinate system of $frame_{n-1}$. We make use of the ICP implementation provided by Open3D. We use $maxcorrespondencedistance = 5000$, $relativefitness = 1e-07$, $relativermse = 1e-07$ and $maxiterations = 50$ as hyperparameters of the ICP algorithm. A more careful selection of these parameters may lead to better results in the later steps of fruit counting.

$$T_{N\to 0} = \prod_{n=0}^{n=N} T_{n\to n-1} \tag{5}$$

By keeping track of the total transform between $frame_n$ and $frame_0$, all detections can be localised relative to the start position of the camera using Equation 5 where $T_{n\to n-1}$ is the transform between $frame_n$ and $frame_{n-1}$.

We use a homogeneous definition of the coordinates system thus $T_{n\to n-1}$ will be defined as:

$$T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The 4×4 homogeneous transformation matrix consists of a 3×3 rotation matrix (R_{ij}) in the top-left, a translation vector (t_x, t_y, t_z) in the top-right, and a bottom row of $[0, 0, 0, 1]$ to maintain homogeneity.

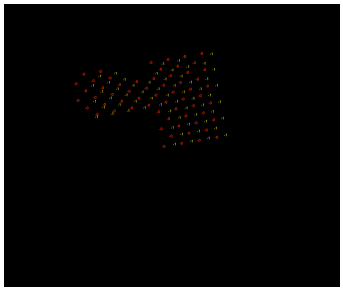


Fig. 8: Centroids of the calibration squares in the unified coordinate system. Colour and number represent the frame number in which the squares were imaged.

$$c_0 = T_{n\to 0} \times c_n \tag{6}$$

Application of Equation 6 to the centroid $c_n \in C_n$ of each detection in frame n, maps it back to the coordinate system of the first image frame, $c_0 \in C_0$.

From figure Figure 8, we can see that there is significant drift in the unified coordinate system introduced between frames. This is likely due to improper selection of hyperparameters for the ICP algorithm as previously mention. The multiplication in Equation 5, means that any error in the estimated transform between two frames will propagate and compound over time.

Once all detections have been localised into a single coordinate system, we can then group duplicate detections using agglomerative clustering to form a linkage matrix followed by running an 'fcluster' algorithm to return the flattened clusters from the hierarchical clustering defined by the given linkage matrix.

We make use of the Ward variance minimization algorithm Equation 7 to compute the distance between clusters. The distance between a newly formed cluster u, which consists of clusters s and t, and another cluster v is calculated as:

$$d(u,v) = \sqrt{\frac{|v|+|s|}{L}d(v,s)^2 + \frac{|v|+|t|}{L}d(v,t)^2 - \frac{|v|}{L}d(s,t)^2} \tag{7}$$

where $L = |v| + |s| + |t|$.

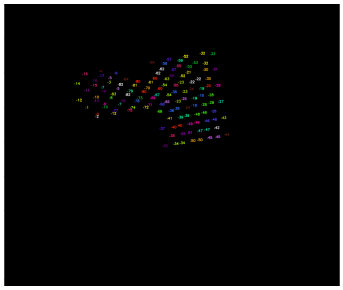


Fig. 9: Centroids of the calibration squares in the unified coordinate system. Colour and number represent the estimated unique grouping of the detections.

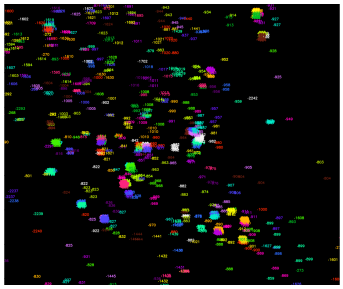


Fig. 10: Centroids of the detected pears in the unified coordinate system. Colour and number represent the estimated unique cluster IDs of the detections. Figure is not to scale.

Figure 9 shows how the application of the agglomerative clustering groups the detections into assumed unique objects. Due to the cumulative error in the calculation of $T_{n\to 0}$, the grouping is not perfect.

Estimating the total number of fruit on a single side of a row of trees within the orchard is then just a case of counting the total number of unique detections for a video of that row.

B. Results

Despite the cumulative error in the calculation of $T_{n\to 0}$, Figure 10 shows that we have largely been successful in grouping the detections of pears in one of the rows of trees into unique pears.

†We do not have a ground truth total count for the unique pears in each of these rows, however each row should have approximately 15000 pears per row. Given that we only image a single side of the row per video, we can assume that we will only see approximately half of the pears. Thus we expect to see approximately 7500 pears on each side of each row.

VIII. Fruit Sizing

A. Method

In order to determine the size of a fruit we can make use of the segmentation mask provided by the spatial-spectral segmentation algorithm and the point cloud generated by DAv2.

By masking the point cloud with the segmentation mask of a detected fruit, we can then estimate the height and width by taking the four points within the mask which are furthest left, right up and down in the imaging plane and computing $width = right - left$ and $height = up - down$.

This method does not account for the rotation of the fruit around the Z axis (orthogonal to the imaging plane) but it is easy to compute and most pears grow approximately vertically due to the increase in mass in the lower half of the fruit and apples are approximately spheres, so their rotation does not matter.

1) Occlusion: One of the main challenges with estimating the size of a detected object is occlusion. If an obstacle, such as a leaf, partially blocks the view of the object from the camera, then its measured extent will not be correct.

To try and counter this effect, we make use of an aspect ratio threshold for the pears. Pears are approximately twice as tall as they are wide, thus any detected pear which does not match this aspect threshold is likely to be either a False-Positve or an Occluded detection.

We make use of an aspect ratio threshold Equation 8 function to filter out the detections which are likely to be of occluded pears. I.e. a detection is assumed to be a partial detection if the height is less than 1.7 times the width of the detection.

$$Fulldetection = \begin{cases} True & \text{if } H >= 1.7W \\ False & \text{else} \end{cases} \tag{8}$$

B. Results

Table VI shows the distribution of pear sizes measured for the sides of the tree rows tested. Whilst the mean sizes are reasonable, there is a large standard deviation on the measurement. This is due to both error in the measurement system, from the calibration method as well as DAv2 itself, as well as the inclusion of obviously under and over sized detections into the calculation.

Pears have an average size of (7.62cm, 11.43cm) [47]. We use a threshold of 33% on either side of these values to define pears which are of the expected size.

The largest minority of detected pears are undersized. This could point to error in the size estimation method, but it could also be due to incorrect selection of the sizing bands. The expected pear size was scrapped from the web and isn't necessarily for the same variety of pears being imaged and a threshold of 33% was some what arbitrarily chosen.

In future, we will need to gather ground truth sizing results to compare against. At the very least, these should include the mean and standard deviation of sizes for each row of pears. We could also make use of the size grading thresholds used by farmers, but these are not known at the time of writing.

Figure 11 shows the distribution of pear sizes measured for each of the rows. All rows are largely normally distributed for both the width and length of pears as is to be expected. However, there are clearly some outliers in the detections, with some detections clearly being too small to be correct and others which are far too large. The very small detections are likely to be of occluded pears. The approach for filtering out

Block ID	Side	Total Detections	Total Unique Detections	Estimated Counting Accuracy †
7-157	Left	2616	1418	18.91%
7-157	Right	6589	2725	36.33%
7-160	Left	7038	2914	38.85%
7-160	Right	19376	3694	49.25%
Combined	-	35619	10751	35.84%

TABLE V: Pear detection statistics by block ID and side of row

Block ID	Side	Size Distribution (%)			Average Width (cm)	Average Length (cm)
		$W < 5.1cm$ $L < 7.6cm$	$5.1cm \leq W < 10.2cm$ $7.6cm \leq L < 15.2cm$	$W > 10.2cm$ $L > 15.2cm$		
157	Left	56.91%	28.18%	18.78%	4.88 ± 2.04	10.75 ± 4.33
157	Right	44.01%	35.33%	23.55%	5.54 ± 1.99	11.94 ± 3.98
7-160	Left	62.57%	27.19%	12.10%	4.68 ± 1.89	10.09 ± 4.05
7-160	Right	13.11%	54.87%	33.71%	6.88 ± 1.56	14.04 ± 2.96
Combined	-	41.59%	38.08%	22.64%	5.62 ± 2.05	11.88 ± 4.08

TABLE VI: Pear Size Distribution by Block ID and Side of Row

partial detections is very naive and does not take into account the absolute size of the detected pears, only caring for the ratio of height to width. The exceptionally large detections may come from poor segmentation mask generation as seen in Figure 6d where False Positive spectral detections have led to a non-pear image segment (a stick) being classified as a pear. Minimum object dimensions may help remove FP detections like these, but the best solution will be further improvements of the spatial-spectral detection model.

IX. Conclusion

We have presented a dataset, benchmarking framework and system for leveraging hyperspectral images for the purpose of detecting, segmenting, counting and sizing fruit in an orchard.

The dataset (livingoptics.com/huggingface) is made public in the hopes that others will develop their own hyperspectral algorithms and drive the state-of-the-art forward for hyperspectral detection and segmentation.

The results from combining a spectral classification algorithm with an RGB segmentation algorithm are promising and demonstrate the value of hyperspectral data in the challenging domain of real-world agriculture. Our proposed method, 1D-ConvNet + SAM2 combination, outperformed the RGB only method (YOLOv8s). The main improvement comes in the form of a nearly 25% improvement in Recall. YOLOv8s struggled to detect more than half of all fruit due to relying solely on shape and the minimal colour information available in RGB images. Given the dataset contains high levels of occlusion, shadowing and motion blur this information is likely to be insufficient in most instances. In contrast, the Spatial-Spectral approach allows for the detection of more than 70% of all fruit. Due to the spectral detection being a point-wise algorithm, only a minimal amount of a fruit needs to be un-occluded for it to be spectrally detected. From there, any spatial-segment with a sufficient proportion of spectral detections within it can be assigned a class. This means that partially occluded fruit which may not have sufficient shape information for YOLOv8s to detect and segment can still be detected and segmented.

On top of the fruit detections, we have developed a two stage process for 1) uniquely tracking each individual detected fruit across a video and then 2) estimating the size of those unique detections. This highlights how a spatial-spectral component can replace a spatial (RGB) only component in a complex computer vision workflow and how the improved performance of the spatial-spectral component can benefit downstream tasks.

X. Future Work

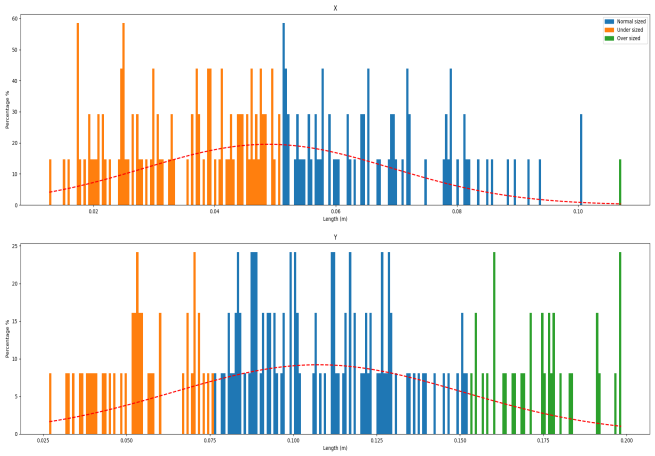
This study highlights the potential for enhancing spatial-spectral detection, segmentation, counting and sizing methodologies in HSI systems. However, several areas for improvement have been identified, which can serve as a foundation for future work.

A. Spectral detection improvements

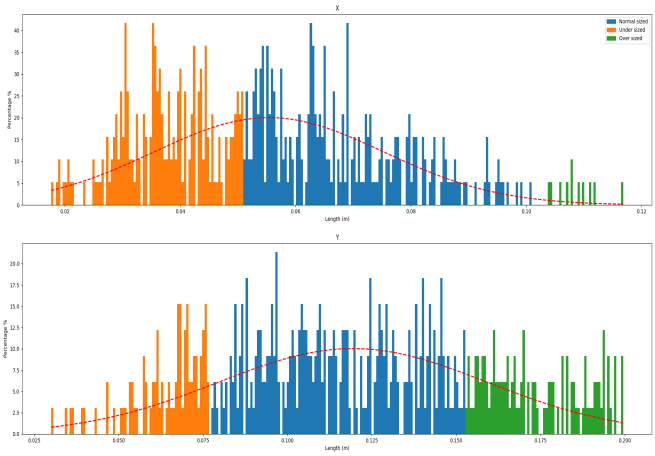
Future efforts will focus on refining the spectral detection algorithm to enhance accuracy and robustness. A broader architectural and hyperparameter search is a priority, aiming to identify optimal model structures that balance complexity and performance. Utilising Bayesian Optimised Hyperband [50] for this purpose would reduce the time taken for this by eliminating poor performing areas of the search space earlier in the search process. Exploring alternative loss functions tailored for hyperspectral data will also be essential, particularly to address imbalanced datasets and improve convergence. Enhancing data balancing strategies, such as employing advanced oversampling or synthetic data generation methods, could improve model reliability. Increasing the amount of training data is another key avenue for improving generalisation and reducing overfitting, potentially through augmentation or the inclusion of more real-world data. Additionally, postprocessing techniques could be developed to refine detections and mitigate False-Positves and negatives.

B. Object segmentation

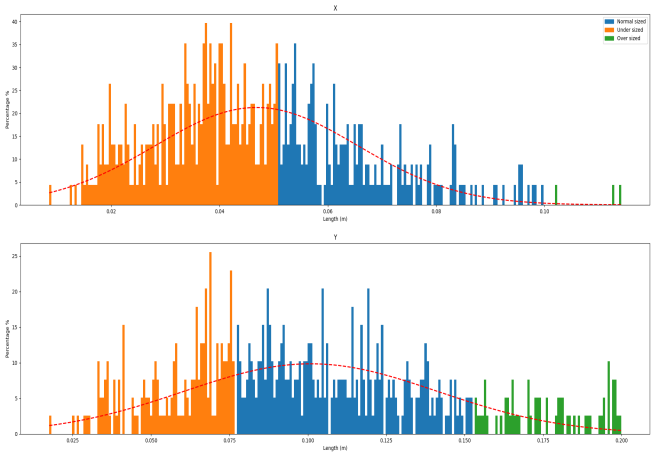
Currently, segmentation relies on off-the-shelf third-party models. Fine-tuning these models on domain-specific data is a logical next step to improve accuracy and efficiency.



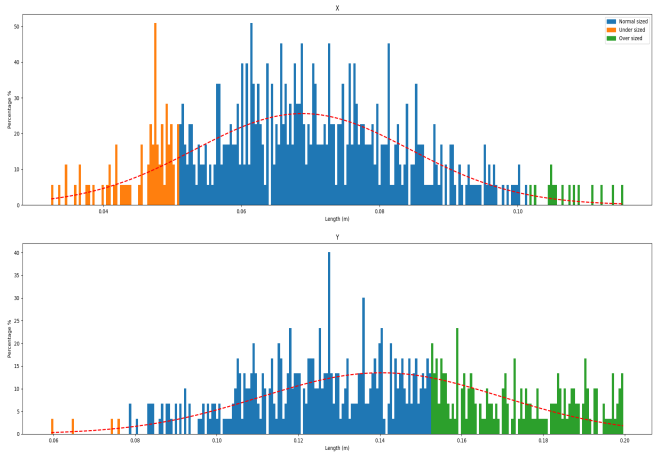
(a) Block 157, left



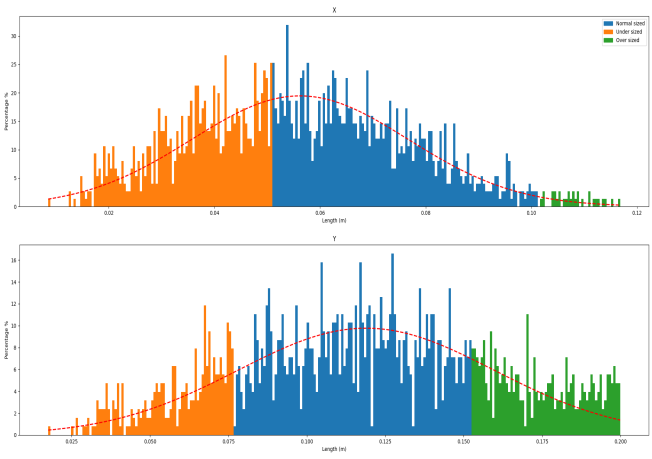
(b) Block 157, right



(c) Block 160, left



(d) Block 160, right



(e) Combined

Fig. 11: Distribution of measured pear sizes. The top subplot in each sub-figure (a - e) is for the pear width (X), the bottom for the pear length (Y).

However, long-term goals include developing a custom RGB segmentation algorithm optimised for the specific requirements of HSI tasks. This bespoke model would prioritise both speed and accuracy, leveraging innovations in deep learning

to outperform generic solutions.

Additionally, a joint spectral and RGB encoder could be developed to jointly leverage both modalities directly, rather than in the two stage approach we have taken in this work where the spectral and RGB data are treated separately.

C. Monocular depth

Depth estimation also relies on an off-the-shelf third-party monocular depth model. Fine-tuning this system to align with the dataset's characteristics could enhance performance. In addition, designing a custom monocular depth estimation algorithm tailored to HSI applications would provide better control over accuracy and computational efficiency. As with the segmentation algorithm, utilising the spectral data directly in a monocular depth algorithm may also offer improvements in performance by providing rich key points around which to build feature embeddings.

Another critical area is improving the depth calibration process to ensure precise unit conversions, as calibration inaccuracies can lead to significant measurement errors. Automating this calibration process and rigorously testing it across various scenarios will be essential for robust system deployment.

D. Tracking

Tracking improvements will focus on unifying the coordinate system used in hyperspectral analysis. Optimisation of the hyperparameters used in Iterative Closest Point (ICP) will improve the alignment of repeat detections across multiple frames, providing more accurate object tracking.

For faster and more efficient processing, replacing ICP with a non-point cloud based approach, such as Scale-Invariant Feature Transform combined with the computation of a homography, could offer significant speed-ups.

One major draw back of the current tracking algorithm is that it assumes a static scene and a moving camera. Further work would be needed to account for object movement within the scene.

E. Sizing

The current sizing approach, based on bounding box dimensions, can be further refined to account for object rotation. Future work will include developing algorithms to determine the maximum length and width of objects by analysing the mask, irrespective of orientation. This approach will ensure that size estimates are more precise and less affected by object positioning.

Capturing ground truth fruit sizes - or at least mean and standard deviation values, would allow for a more empirical analysis of the sizing results

F. Yield estimation

Getting ground truth counts of the unique fruit would improve the quality of the analysis in this work.

Furthermore, by combining the counts from the tracking system with the sizing data an estimate of the total volume of fruit can be made. By adding an additional layer of spectral metrics to estimate the quality and health of the detected fruit a more granular yield estimate could be made. For example, counts of healthy, damaged or diseased fruit could be generated.

XI. Appendix A

Preliminary experiments with the Random Forest classifier demonstrated the necessity of forming a background class in the training data. This is due to two factors:

1) Random forests must assign a class to all inputs - leading to background spectra being assigned foreground classes if no background class is utilised.
2) The subtle difference between foreground class spectra and background spectra mean that the confidence score of the random forest is not wholly reliable for distinguishing between some foreground and background spectra. Therefore we cannot easily suppress background spectra classified as foreground classes.

The background class is formed of all spectra which do not lie inside a mask from the foreground labelling.

The background class contains many more times the number of spectra that are found in all foreground classes combined - for the training set 1,855,755 background spectra vs 51,285 foreground spectra. This biases both any algorithm trained on the data to preferentially select the background class over all other classes if no attempt is made to either balance the training data or the training loss.

In the case of the random forest, we have no notion of epochs of training or data batching, therefore if we want to make use of all of the training data, we cannot easily remove background spectra to balance the number of foreground vs background spectra.

Instead we have two possible directions to take to achieve a balanced training set:

1) Class weighting
2) Foreground data duplication

1) Class weighting: We can provide a set of weights to the random forest training algorithm to reduce the importance of correctly classifying the background class vs the foreground class. These weights are calculated as being inversely proportional to the frequency of the class. This can be done at either the dataset level or at the subsample level and is directly supported in Scikit-learn by passing either 'None', 'balanced' or 'balanced_subsample' to the 'class_weight' argument of the random forest.

We can take a similar approach when training the 1D convolutional neural network, providing a set of class weights which can be used to adjust the loss for each class so as to attempt to avoid biasing the model towards the background class. We only calculate the weighting once, for the entire dataset, rather than on a batch by batch basis as this would require reinitialising the loss function on each batch which would be prohibitively slow during training. Though such an approach might yield better results by re-balancing the loss each batch.

$$w_c = \frac{N}{n_c} \tag{9}$$

In Equation 9, the weighting associated with a class, $w_c \in W$, is inversely proportional to the frequency of class c. N is the total number of spectra in the training data and n_c is the number of spectra of class c in the training data. We then normalise the weights by the maximum, $W_{norm} = \frac{W}{max(W)}$.

2) Foreground data duplication: We could choose to duplicate a random sample of the foreground spectra so that the number of background and foreground spectra match as well as providing balancing across classes.

However, without a data augmentation method, this duplication isn't helpful for training the random forest as duplicated spectra will be classified by the same nodes, without the need for additional nodes to distinguish them. As spectral data augmentation is beyond the scope of this work, we do not explore this approach for the random forest.

For the 1D convolutional classifier, directly duplicating the data is a more reasonable approach, even without data augmentation as this simply biases the loss away from the background class by providing more frequent weight updates calculated from the foreground class losses.

There are two ways in which we can duplicate the foreground data - 1) match the total number of foreground spectra to the total number of background spectra 2) match the number of spectra in each foreground class to the number of spectra in the background class.

$$D_f = duplicate(D_f, N_b) \tag{10}$$

Equation 10 uses a function called duplicate which will create copies of D_f, the foreground data, so that N_f, the number of foreground spectra matches N_b the number of background spectra.

$$D_fc = duplicate(D_fc, N_b) \tag{11}$$

Equation 11 uses the duplicate function to match the number of spectra in class c to the number of background spectra N_b. By applying this method to all $c \in C$, all foreground classes will have the same number of samples as the background class.

3) Background data sampling: A third option for dataset balancing is available to the 1D convolutional classifier - background data sampling.

As the convolutional classifier is trained for a number of epochs, we can randomly select a sample of the background data at the start of each epoch such that the training set for each consists of N foreground spectra and N background spectra.

Alternatively, we could select the number of background spectra to be equal to the number of spectra in the largest foreground class.

Background data sampling is a good approach to solving the data imbalance problem as it still allows us to leverage all of the background data whilst ensuring that the average loss per epoch is not overly biased towards the background class.

REFERENCES

[1] Mohamed Hisham Aref, Sanzhar Korganbayev, Ibrahim H Aboughaleb, Abdallah Abdelkader Hussein, Mohamed A Abbass, Ramy Abdlaty, Yasser M Sabry, Paola Saccomandi, and Abou-Bakr M Youssef. Custom hyperspectral imaging system reveals unique spectral signatures of heart, kidney, and liver tissues. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 305:123363, 2024.

[2] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Deep learning for classification of hyperspectral data: A comparative review. *IEEE geoscience and remote sensing magazine*, 7(2):159–173, 2019.

[3] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[4] Ricardo Augusto Borsoi, Tales Imbiriba, and José Carlos Moreira Bermudez. A data dependent multiscale model for hyperspectral unmixing with spectral variability. *IEEE Transactions on Image Processing*, 29:3638–3651, 2020.

[5] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020.

[6] Steve Chappell, Robin Wang, Sam Hornett, and Alex Spanellis. A VIS-NIR hyperspectral video imager. In Lynda E. Busse and Yakov Soskind, editors, *Photonic Instrumentation Engineering XI*, volume PC12893, page PC1289307. International Society for Optics and Photonics, SPIE, 2024.

[7] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[8] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, and Yanfeng Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing*, 7(6):2094–2107, 2014.

[9] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020.

[10] Lucas Drumetz, Jocelyn Chanussot, Christian Jutten, Wing-Kin Ma, and Akira Iwasaki. Spectral variability aware blind hyperspectral image unmixing based on convex geometry. *IEEE Transactions on Image Processing*, 29:4568–4582, 2020.

[11] Rogério P Espíndola and Nelson FF Ebecken. On extending f-measure and g-mean metrics to multi-class problems. *WIT Transactions on Information and Communication Technologies*, 35:25–34, 2005.

[12] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2016.

[13] Salvador Gutiérrez, Alexander Wendel, and James Underwood. Ground based hyperspectral imaging for extensive mango yield estimation. *Computers and Electronics in Agriculture*, 157:126–135, 2019.

[14] Jisoo Ham, Yangchi Chen, Melba M Crawford, and Joydeep Ghosh. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):492–501, 2005.

[15] Nicolai Häni, Pravakar Roy, and Volkan Isler. Apple counting using convolutional neural networks. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2559–2565. IEEE, 2018.

[16] Hong-Ju He, Yuling Wang, Xingqi Ou, Hanjun Ma, Hongjie Liu, and Jianhua Yan. Rapid determination of chemical compositions in chicken flesh by mining hyperspectral data. *Journal of Food Composition and Analysis*, 116:105069, 2023.

[17] Danfeng Hong, Zhu Han, Jing Yao, Lianru Gao, Bing Zhang, Antonio Plaza, and Jocelyn Chanussot. Spectralformer: Rethinking hyperspectral image classification with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2021.

[18] Tien-Heng Hsieh and Jean-Fu Kiang. Comparison of cnn algorithms on hyperspectral image classification in agricultural lands. *Sensors*, 20(6):1734, 2020.

[19] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015(1):258619, 2015.

[20] Garima Jaiswal, Ritu Rani, Harshita Mangotra, and Arun Sharma. Integration of hyperspectral imaging and autoencoders: Benefits, ap-

plications, hyperparameter tunning and challenges. *Computer Science Review*, 50:100584, 2023.

[21] Shahid Karim, Akeel Qadir, Umar Farooq, Muhammad Shakir, and Asif A Laghari. Hyperspectral imaging: a review and trends towards medical imaging. *Current Medical Imaging*, 19(5):417–427, 2023.

[22] Dmitry O Khort, Alexey Kutyrev, Igor Smirnov, Nikita Andriyanov, Rostislav Filippov, Andrey Chilikin, Maxim E Astashev, Elena A Molkova, Ruslan M Sarimov, Tatyana A Matveeva, et al. Enhancing sustainable automated fruit sorting: Hyperspectral analysis and machine learning algorithms. *Sustainability*, 16(22):10084, 2024.

[23] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Hyungtae Lee and Heesung Kwon. Going deeper with contextual cnn for hyperspectral image classification. *IEEE Transactions on Image Processing*, 26(10):4843–4855, 2017.

[25] Jun Li, José M Bioucas-Dias, and Antonio Plaza. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4085–4098, 2010.

[26] Shutao Li, Weiwei Song, Leyuan Fang, Yushi Chen, Pedram Ghamisi, and Jon Atli Benediktsson. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6690–6709, 2019.

[27] Delia Lorente, Nuria Aleixos, JUAN Gómez-Sanchis, Sergio Cubero, Oscar Leonardo García-Navarrete, and José Blasco. Recent advances and applications of hyperspectral imaging for fruit and vegetable quality assessment. *Food and Bioprocess Technology*, 5:1121–1142, 2012.

[28] Bin Luo, Chenghai Yang, Jocelyn Chanussot, and Liangpei Zhang. Crop yield estimation based on unsupervised linear unmixing of multidate hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 51(1):162–173, 2012.

[29] Li Ma, Melba M Crawford, and Jinwen Tian. Local manifold learning-based k-nearest-neighbor for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4099–4109, 2010.

[30] Pengfei Ma, Jiaoli Li, Ying Zhuo, Pu Jiao, and Genda Chen. Coating condition detection and assessment on the steel girder of a bridge through hyperspectral imaging. *Coatings*, 13(6):1008, 2023.

[31] Walter Maldonado Jr and José Carlos Barbosa. Automatic green fruit counting in orange trees using digital images. *Computers and Electronics in Agriculture*, 127:572–581, 2016.

[32] Farid Melgani and Lorenzo Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on geoscience and remote sensing*, 42(8):1778–1790, 2004.

[33] María Antonia Diaz Mendoza, Emiro De La Hoz Franco, and Jorge Eliecer Gómez Gómez. Technologies for the preservation of cultural heritage—a systematic review of the literature. *Sustainability*, 15(2):1059, 2023.

[34] Dedong Min, Jiangsan Zhao, Gernot Bodner, Maratab Ali, Fujun Li, Xinhua Zhang, and Boris Rewald. Early decay detection in fruit by hyperspectral imaging–principles and application potential. *Food Control*, 152:109830, 2023.

[35] Juan C Miranda, Jordi Gené-Mola, Manuela Zude-Sasse, Nikos Tsoulias, Alexandre Escolà, Jaume Arnó, Joan R Rosell-Polo, Ricardo Sanz-Cortiella, José A Martínez-Casasnovas, and Eduard Gregorio. Fruit sizing using ai: a review of methods and challenges. *Postharvest Biology and Technology*, 206:112587, 2023.

[36] Hiroshi Okamoto and Won Suk Lee. Green citrus detection using hyperspectral imaging. *Computers and electronics in agriculture*, 66(2):201–208, 2009.

[37] Yuan-Yuan Pu, Yao-Ze Feng, and Da-Wen Sun. Recent progress of hyperspectral imaging on quality and safety inspection of fruits and vegetables: a review. *Comprehensive Reviews in Food Science and Food Safety*, 14(2):176–188, 2015.

[38] Waqar Shahid Qureshi, Alison Payne, KB Walsh, R Linker, O Cohen, and MN Dailey. Machine vision for counting fruit on mango tree canopies. *Precision Agriculture*, 18:224–244, 2017.

[39] P Rajkumar, N Wang, G EImasry, GSV Raghavan, and Y Gariepy. Studies on banana fruit quality and maturity stages using hyperspectral imaging. *Journal of food engineering*, 108(1):194–200, 2012.

[40] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint*, 2024.

[41] Omri Safren, Victor Alchanatis, Viacheslav Ostrovsky, and Ofer Levi. Detection of green apples in hyperspectral images of apple-tree foliage using machine vision. *Transactions of the ASABE*, 50(6):2303–2313, 2007.

[42] Hui Shao, Xingyun Li, Fuyu Wang, Long Sun, Cheng Wang, and Yuxia Hu. Feasibility study on fruit parameter estimation based on hyperspectral lidar point cloud. *Journal of Food Measurement and Characterization*, 18(8):7185–7197, 2024.

[43] John Stamford, Seher Bahar Aciksoz, and Tracy Lawson. Remote sensing techniques: hyperspectral imaging and data analysis. In *Photosynthesis: Methods and Protocols*, pages 373–390. Springer, 2024.

[44] Hao Sun, Xiangtao Zheng, Xiaoqiang Lu, and Siyuan Wu. Spectral–spatial attention network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3232–3245, 2019.

[45] R Wan Nurazwin Syazwani, H Muhammad Asraf, MA Megat Syahirul Amin, and KA Nur Dalila. Automated image identification, detection and fruit counting of top-view pineapple crown using machine learning. *Alexandria Engineering Journal*, 61(2):1265–1276, 2022.

[46] Nutcha Taneepanichskul, Helen C Hailes, and Mark Miodownik. Automatic identification and classification of compostable and biodegradable plastics using hyperspectral imaging. *Frontiers in Sustainability*, 4:1125954, 2023.

[47] Shuncy Gardening Team. How big is a pear?, 2024. Accessed: 2024-12-06.

[48] Radhesyam Vaddi and Prabukumar Manoharan. Hyperspectral image classification using cnn with spectral and spatial features integration. *Infrared Physics & Technology*, 107:103296, 2020.

[49] Juan Pablo Vasconez, Jose Delpiano, Stavros Vougioukas, and F Auat Cheein. Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation. *Computers and Electronics in Agriculture*, 173:105348, 2020.

[50] Jiazhuo Wang, Jason Xu, and Xuejun Wang. Combination of hyperband and bayesian optimization for hyperparameter optimization in deep learning, 2018.

[51] Nan-Nan Wang, Da-Wen Sun, Yi-Chao Yang, Hongbin Pu, and Zhiwei Zhu. Recent advances in the application of hyperspectral imaging for evaluating fruit quality. *Food analytical methods*, 9:178–191, 2016.

[52] Sida Wang and Christopher Manning. Fast dropout training. In *international conference on machine learning*, pages 118–126. PMLR, 2013.

[53] Xuan Wei, Fei Liu, Zhengjun Qiu, Yongni Shao, and Yong He. Ripeness classification of astringent persimmon using hyperspectral imaging technique. *Food and Bioprocess Technology*, 7:1371–1380, 2014.

[54] Guantao Xuan, Chong Gao, and Yuanyuan Shao. Spectral and image analysis of hyperspectral data for internal and external quality assessment of peach fruit. *Spectrochimica acta part A: molecular and biomolecular spectroscopy*, 272:121016, 2022.

[55] Ce Yang, Won Suk Lee, and Paul Gader. Hyperspectral band selection for detecting different blueberry fruit maturity stages. *Computers and Electronics in Agriculture*, 109:23–31, 2014.

[56] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[57] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

[58] Wei Yang, Tyler Nigon, Ziyuan Hao, Gabriel Dias Paiao, Fabián G Fernández, David Mulla, and Ce Yang. Estimation of corn yield based on hyperspectral imagery and convolutional neural network. *Computers and Electronics in Agriculture*, 184:106092, 2021.

[59] Chunyan Yu, Meng Zhao, Meiping Song, Yulei Wang, Fang Li, Rui Han, and Chein-I Chang. Hyperspectral image classification method based on cnn architecture embedding with hashing semantic feature. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(6):1866–1881, 2019.

[60] Jaime Zabalza, Jinchang Ren, Jiangbin Zheng, Huimin Zhao, Chunmei Qing, Zhijing Yang, Peijun Du, and Stephen Marshall. Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. *Neurocomputing*, 185:1–10, 2016.

[61] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything, 2023.

[62] Xuhui Zhao, Thomas F Burks, Jianwei Qin, and Mark A Ritenour. Effect of fruit harvest time on citrus canker detection using hyperspectral reflectance imaging. *Sensing and Instrumentation for Food Quality and Safety*, 4:126–135, 2010.