Computers and Electronics in Agriculture 234 (2025) 110246

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

A novel self-supervised method for in-field occluded apple ripeness determination

Ziang Zhao [a],*, Yulia Hicks [a], Xianfang Sun [b], Benjamin J. McGuinness [c], Hin S. Lim [c]

[a] School of Engineering, Cardiff University, Cardiff, CF243AA, Wales, United Kingdom
[b] School of Computer Science and Informatics, Cardiff University, Cardiff, CF244AG, Wales, United Kingdom
[c] School of Engineering, University of Waikato, Hamilton, 3240, Waikato, New Zealand

ARTICLE INFO

Keywords:
In-field
Self-supervised
Unlabelled
Ripeness
Occlusion

ABSTRACT

The full view of the apples in the orchard is often obscured by leaves and trunks, making it challenging to accurately determine their ripeness, whilst it is an essential yet difficult task for apple-harvesting robots. Within this context, we propose a novel method to address two critical challenges: ripeness determination and in-field occlusion. The proposed method is trained in a self-supervised manner on a dataset consisting of less than 1% labelled images and the rest of unlabelled images. It is made up of three key parts: a reconstructor, a feature extractor, and a predictor. The reconstructor is designed to reconstruct the missing parts of occluded apples. The feature extractor is introduced to learn ripeness-related features from the vast number of unlabelled images. Unlike the previous approaches classifying the fruit ripeness into several discrete categories, the predictor uses the learned features to generate a continuous ripeness score in the range between 0.0 and 1.0, thus eliminating the need to subjectively pre-define ripeness stages and offering end-users the flexibility to make their own decisions.

Experimental results comparing our method to another method with different settings show that our method achieves the best Structural Similarity Index Measure (SSIM) of 0.75 and the second-best Peak-Signal-to-Noise Ratio (PSNR) of 25.36 for reconstructing missing apple parts, whilst using the fewest 86.3M parameters. Besides, our method outperforms 15 other self-supervised methods and even a supervised method in the ripeness score prediction, with the smallest score 0.0127 for fully unripe and the highest score 0.8933 for fully ripe apples. The results demonstrate the potential of our method to be incorporated with in-field robotic systems, enabling them to assess ripeness for selective harvesting effectively. It is helpful to monitor the overall ripeness of large orchards digitally, aid the decision-making processes and advance the goals of smart and precision agriculture.

1. Introduction

1.1. Background

Apples are one of the most popular fruits globally, cherished for their taste and nutritional value. Food and Agriculture Organization of the United Nations (FAO) reports that global apple production has steadily increased since 2017 (FAO, 2024). This growth has been supported by advancements in agricultural technology, which have contributed to increased mechanization in apple production. Despite these advancements, apple harvesting remains a labour-intensive and time-consuming process, and it is facing the growing challenge of labour shortages recently.

Significant research efforts across the world have been devoted to the development of fruit-harvesting robots over the past few years. Silwal et al. (2017) designed a cost-effective robotic apple harvester that successfully picked 84% of the apples in a commercial orchard. Kang et al. (2020) developed an apple harvesting system, with a lightweight detection network with PointNet for pose estimation. Zhang et al. (2021) further proposed an apple harvesting prototype that integrates a fruit detection model, a three-degree-of-freedom manipulator, and a vacuum end-effector. Bu et al. (2022) evaluated a robotic apple harvester and found that the “horizontal pull with bending” motion outperformed the anthropomorphic motion in success rate and speed while avoiding stem-pulling and bruising. Besides, a spatio-temporal model was introduced to detect in-field pineapple, achieving a high

* Corresponding author.
E-mail address: zhaoz60@cardiff.ac.uk (Z. Zhao).

https://doi.org/10.1016/j.compag.2025.110246
Received 16 December 2024; Received in revised form 21 February 2025; Accepted 3 March 2025
Available online 19 March 2025

Fig. 1. Apples with distinct ripeness difference can appear simultaneously.

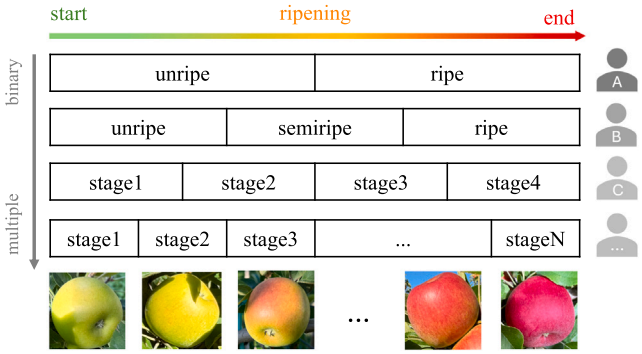


Fig. 2. Different users have different criteria for apple ripeness.

detection accuracy for pineapple-picking robot (Meng et al., 2023). For pitaya fruit harvesting, Li et al. (2024) improved the YOLOv5s model to work in both day and night environments, and deployed it on a mobile device. Jangali et al. (2024) presented a multi-purpose robotic end effector with vacuum suction and rotation, achieving 66.1% successful rate in apple thinning and showing potential for harvesting. Lammers et al. (2024) developed a dual-arm robotic apple harvesting system with improved perception and coordination algorithms. Chen et al. (2024) proposed an approach to develop fruit-picking robots by proposing vision algorithms for efficient locomotion, self-positioning, and dynamic harvesting.

However, most of above fruit-harvesting robots do not consider fruit ripeness during operation, meaning that all fruit are harvested at the same time. In the context of apple precision agriculture, variations in apple ripening times exist both among trees within the same orchard and even among apples on the same tree, as illustrated in Fig. 1. The differences in ripening times are influenced by a combination of environmental conditions, biological traits, and human interventions. This lack of selectivity can lead to reduced apple market value and the need for post-harvest sorting. Therefore, it is necessary for the harvesting robots to adopt a selective harvesting approach that focuses only on ripe apples.

1.2. Ripeness determination

Ripeness determination is the first challenge in this work. In the past decades, researchers have developed many methods to identify the ripeness stage of apples and other fruits. These methods can be broadly categorized into destructive and non-destructive approaches. Destructive approaches rely on analysing fruit's internal attributes such as titratable acidity, soluble solids content, and total soluble solids. Qin et al. (2009) found that spectral scattering, either across all wavelengths or selected ones, provides accurate predictions of apple ripeness. Liu et al. (2016) analysed changes in colour, soluble sugars, organic acids, anthocyanins, and aroma components during apple ripening using liquid chromatography and gas chromatography-mass spectrometry. Das et al. (2016) measured ultra-violet fluorescence from chlorophyll in apple skin across different varieties during ripening and correlated it with destructive firmness tests to assess ripeness.

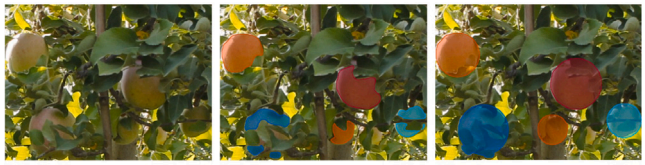
In contrast, non-destructive approaches have recently gained more attention due to they are cost-efficient and do not damage the fruit. These approaches often take imaging as input data. Liu et al. (2015) used multispectral imaging with 19 wavelengths to predict the ripeness of tomatoes. Besides, it is noted that deep learning has recently emerged as a non-destructive approach for classifying fruit ripeness stages using RGB images as input data. Saranya et al. (2021) proposed a convolutional neural network (CNN) to classify bananas into four ripeness stages and compared its performance with state-of-the-art CNNs using transfer learning. Suharjito et al. (2021) developed a mobile application for classifying the ripeness levels of oil palm fresh fruit bunches, utilizing lightweight CNN MobileNets (Howard et al., 2017). Several studies have explored the use of vanilla and customized CNNs to predict different fruit ripeness stages. DenseNet (Huang et al., 2017) was applied to assess ripeness in mulberries (Miraei Ashtiani et al., 2021). VGG (Simonyan and Zisserman, 2015) was utilized to predict the ripeness levels of grapes (Ramos et al., 2021).

Moreover, some studies have incorporated fruit ripeness classification into detection and segmentation tasks. For instance, Xiao et al. (2021) employed a two-step approach to detect apples, first using Fast-RCNN (Girshick, 2015) to predict apple locations, followed by classifying them into 3 ripeness stages. Zhao et al. (2023b) introduced a novel one-stage instance segmentation model that directly segments peaches and classifies them into 3 ripeness stages. Wang et al. (2023) proposed a feature augmentation network with decoupled heads to segment strawberries and classify them into 2 ripeness stages. Wang et al. (2024) proposed a new class balance method and a YOLO-based network, for segmenting tomatoes and classifying them into 3 ripeness stages. These studies have demonstrated the success of deep learning models in classifying fruit ripeness stages.

However, a key limitation of these studies is that they rely on the pre-defined number of ripeness stages during image labelling, which may not align with real-world decision-making processes. Specifically, simulating the decisions of end-users (e.g., farmers, orchard managers) is difficult, as their criteria for ripeness often vary based on individual goals. For example, some managers targeting long-distance markets may prefer to harvest apples early before fully ripe, while some may like to harvest apples only when fully ripe.

Determining apple ripeness from images is usually a subjective and challenging task. Fig. 2 shows that the definitions of "ripe" can be different among different users, ranging from binary classifications to more granular multi-category classifications. Binary and three-category classifications are the most commonly considered by previous research. However, extending these models to finer classifications, such as five categories, requires re-labelling the images and retraining the model, which introduces unnecessary effort. To solve this, we regard ripeness determination as a regression task rather than a multi-category classification task.

It is noted that regardless of the number of ripeness stages defined by the users, the fully unripe and fully ripe apples will always remain in the first and last categories, respectively. Based on this, we proposed a self-supervised method which takes few images of fully unripe and fully ripe apples as labels, learns from a large number of unlabelled images, and generates ripeness scores as output.



(a) Image (b) Modal Masks (c) Amodal Masks

Fig. 3. Example of modal and amodal masks (Gené-Mola et al., 2023).

1.3. In-field occlusion

In-field occlusion is the second challenge in this work. Since most of these robots heavily depend on visual perception for fruit identification and localization, occlusion significantly impacts their decision-making process. As shown in Fig. 1, apples are often easily occluded by leaves. Moreover, occlusion can also result in recognition failures, requiring manual leaf removal prior to picking (Van Herck et al., 2020).

Some of previous research has considered the occlusion when training the detection and segmentation models. Tian et al. (2019) introduced a YOLO-based model specifically designed for detecting apples at different growth stages in orchards and mitigated apple overlap and occlusion to some extent. Zheng et al. (2021) proposed a CNN-based vision algorithm for mango instance segmentation and picking point localization, considering occlusion, overlap, and variations in object scale. Wang et al. (2024) replaced the network's complete-IoU regression loss function with the weighted-IoU loss function to address tomato fruit and leaf occlusion. Chen et al. (2023) proposed a YOLO-based lightweight 4-class occlusion detection method for Camellia oleifera fruit, introducing a clustering algorithm to select the target dataset. Similarly, Du et al. (2023) proposed a detection model to locate ripe ground-planted strawberries of 4 different occlusion categories.

Furthermore, some researchers proposed to estimate the shape of partially occluded fruits by means of amodal instance segmentation, which aims to predict the shape of each object of interest in an image (Li and Malik, 2016). Gené-Mola et al. (2023) implemented an amodal segmentation model with an end-to-end CNN for accurate Fiji apple detection and sizing, predicting complete shapes (visible and occluded regions) and achieving robust diameter estimation. The examples of modal and amodal masks are shown in Fig. 3. Kim et al. (2023) employed an amodal segmentation approach using a reconstruction network to perform cucumber occlusion recovery, achieving high accuracy and speed. Besides, some researchers introduced mathematical methods to estimate the shape of the target fruit. Sun et al. (2024) proposed an active deep sensing method to handle occlusions in clustered and single fruit scenarios, utilizing a deep network to predict optimal observation positions, and guiding robots to avoid the occlusion. Liang et al. (2024) mitigated the challenge of fruit occlusion in complex environments by leveraging approximately spherical fruit shape priors for improved segmentation and localization, enabling effective occlusion-aware solutions without reliance on additional data or equipment.

However, all of the above research limits the addressed problem to either classifying the occlusion categories or estimating the shape of the occluded fruit. Taking a step forward, we propose a self-supervised method to reconstruct the details of the occluded parts of the fruits.

1.4. Self-supervised learning

Self-supervised learning is a promising path to advance machine learning, which can learn from a large number of unlabelled data (Balestriero et al., 2023). There have been some efforts in applying self-supervised learning in the agricultural sector, such as cherry maturity detection (Gai et al., 2023), leaf disease identification (Zhao et al., 2023a), and crop anomaly detection (Choi et al., 2024). Different from them, we adopt self-supervised learning to extract features related to apple ripeness and address the in-field occlusion problem.

Fig. 4. Left: The apple orchard. **Right:** Samples of apple images.

1.5. Contributions

To address the mentioned two challenges, this paper is devoted to in-field occluded apple ripeness determination with few labelled images and vast unlabelled images. Specifically, we proposed a self-supervised method, which consists of three parts: a reconstructor, a feature extractor, and a ripeness score predictor. To the best of the authors' knowledge, our method is novel and has not been explored in prior research.

The main contributions of our proposed method are summarized as follows:

1. The reconstructor is trained on unlabelled "complete" apple images to learn, and then apply the acquired knowledge to reconstruct the details of missing parts for "incomplete" apples.
2. The feature extractor is designed to capture ripeness-related features from a large number of apple images, of which less than 1% images are labelled.
3. The predictor eliminates the need for subjectively pre-defining the number of ripeness stages, instead, it generates a continuous "ripeness score" between 0.0 and 1.0, allowing end-users to make decisions based on their criteria.

2. Dataset

2.1. Image collection

We captured a number of 2530 apple images (4032 × 3024 pixels) with a mobile phone in a large Jazz apple orchard located near Hawke's Bay, New Zealand. The overview of the orchard and samples of the apple images are presented in Fig. 4. The collection took several weeks from February to March, and encompassed the complete apple ripening progress from fully unripe to fully ripe.

There were no specific requirements for the image collection. All apple images were taken under natural illumination and in real-world production settings, taken from various angles to simulate every possible scenario for the in-field operation of robots. As a result, the apples exhibited variations such as being isolated, in close proximity to each other, and partially obscured by leaves or stalks.

2.2. Image preprocessing

We use YOLO-World (Cheng et al., 2024) to detect the bounding boxes of apples, and then use the boxes as the input of Segment Anything Model (SAM, Kirillov et al., 2023) to perform the apple instance segmentation. The workflow of the process is shown in Fig. 5. The dataset consists of 2530 images, from which 7191 apple instances were detected and segmented following the workflow. From these, we

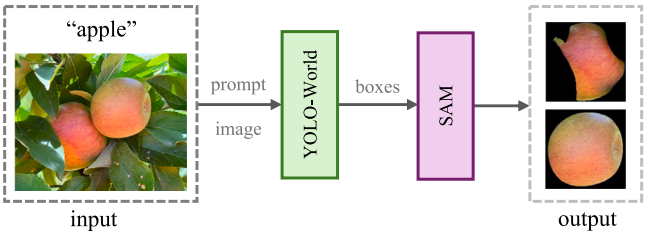


Fig. 5. The workflow of image preprocessing.

Fig. 6. The selected 20 fully unripe and 20 fully ripe apples.

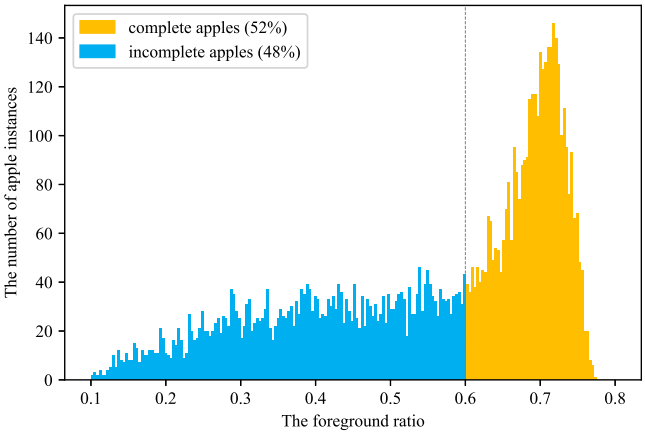


Fig. 7. The Fr distribution of the dataset.

manually selected 20 fully unripe and 20 fully ripe apples under diverse conditions, using them as labelled instances, as illustrated in Fig. 6, while the remaining 7151 apple instances remain unlabelled.

$$\mathrm{Fr} = \frac{N_{apple}}{N_{img}} \tag{1}$$

We compute the foreground ratio Fr of all uniformly resized apple instances using Eq. (1), where N_{apple} represents the number of pixels corresponding to apples, and N_{img} is the total number of pixels in the image.

The distribution of Fr is presented in Fig. 7. Here, we define apples with Fr $\geq$ 0.6 as “complete” apples, as they contain sufficient visual information for analysis. In contrast, apples with Fr $<$ 0.6 are categorized as “incomplete” apples, as substantial portions of the apple are occluded, resulting in limited details.

2.3. Image augmentation

Image augmentation involves applying various transformations to images to artificially increase the size of a dataset and simulate real-world conditions.

For some of self-supervised learning methods, image augmentation is a cornerstone of training strategies. It serves as a key mechanism

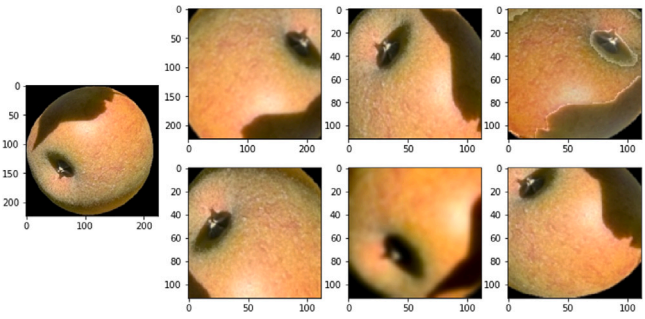


Fig. 8. Left: The original image. **Right:** Examples generated via augmentation.

to manipulate input data, ensuring that the model learns meaningful representations from a large number of unlabelled data.

Based on the collected apple images, we assume that some in-field conditions observed in apples, such as variations in brightness, shadows, viewing angles, and occlusions caused by leaves, branches, or other fruits, can be regarded as forms of ’natural augmentation’. These natural augmentations do not influence the ripeness of the apples, as ripeness is an intrinsic quality independent of external conditions.

In this paper, we incorporate a variety of artificial augmentation methods, including random cropping, random scaling, random flipping, brightness adjustment, colour jittering and Gaussian blur to simulate natural augmentations. For instance, random cropping and flipping mimic the perspectives of images captured from different angles, while Gaussian blur replicates the effect of images taken when the camera is out of focus on the apples. It is noted that gray-scale conversion is not used in our work, as it results in the loss of colour information. The illustration of augmentations is provided in Fig. 8.

By setting different probabilities to each method, we generate a diverse set of variations, enabling the model to robustly learn meaningful features associated with ripeness across different scenarios.

3. Proposed method

3.1. Overview

The overall architecture of our work is shown in Fig. 9. The collected images first undergo a preprocessing stage, including object detection and instance segmentation. Following this, the apple instances are partitioned based on two criteria: (1) whether they are labelled and (2) whether they are complete or incomplete. Complete apples are utilized for feature extraction and reconstruction, and incomplete apples are used for reconstruction. Finally, labelled apples serve as boundaries for projecting the features onto the final ripeness prediction.

Specifically, our proposed method contains three parts: a missing-part reconstructor, a feature extractor and a ripeness score predictor. The framework of our method is illustrated in Fig. 10.

- **Reconstructor**
  The reconstructor is a self-supervised component designed for incomplete apples, which aims to reconstruct missing parts of apples to provide more details.
- **Extractor**
  The extractor also operates within a self-supervised paradigm to learn representations related to ripeness from images. Specifically, it is expected to find a feature space in which every apple is separated by its ripeness, and unripe apples are as far as possible from ripe apples.
- **Predictor**
  The predictor is a simple multi-layer perceptron (MLP), which takes features from the extractor as input and predicts ripeness scores.

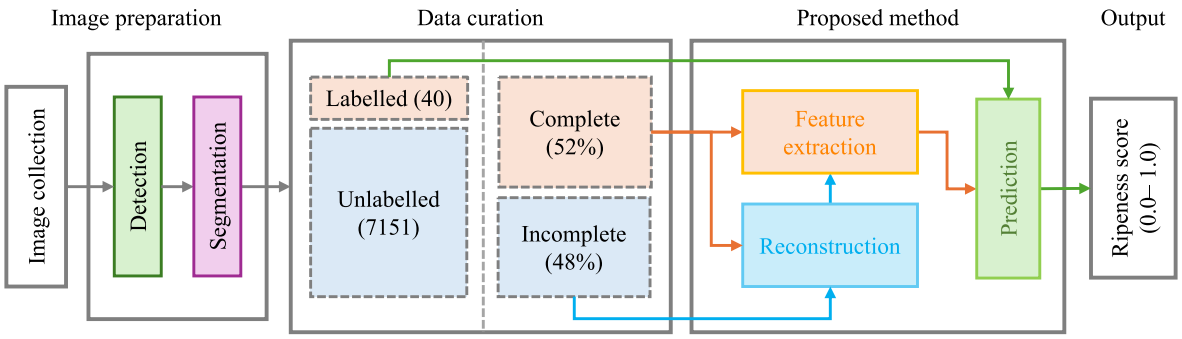


Fig. 9. The overall architecture of our work.

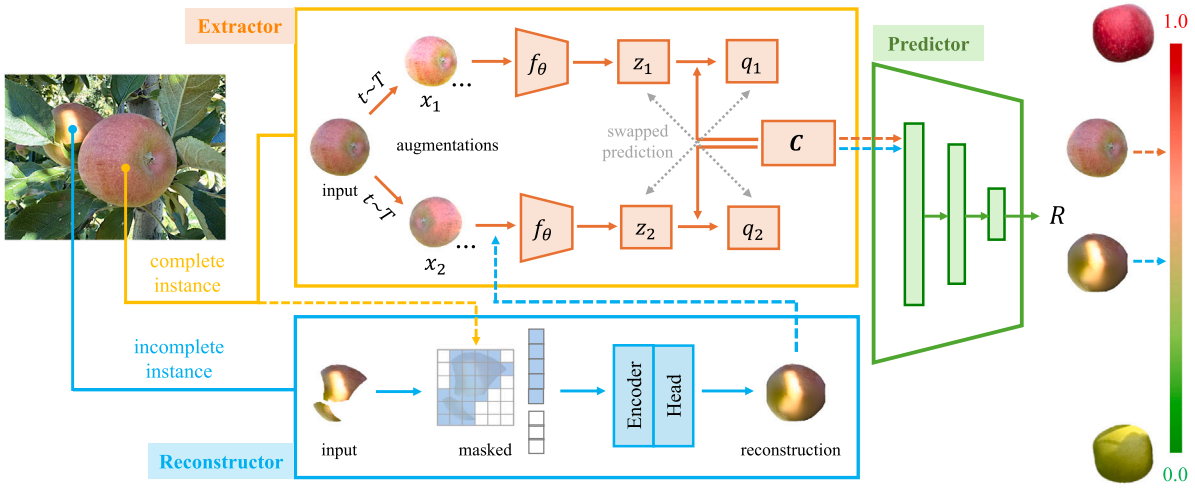


Fig. 10. The framework of our proposed method.

3.2. Reconstructor

The reconstructor is based on 'Masked Image Modelling', which learns by masking portions of the input image and predicting the missing parts. In this context, we consider occlusions caused by leaves or trunks as a kind of natural mask, and the task is to reconstruct these occluded apples.

Specifically, our reconstructor employs the SimMIM (Xie et al., 2022), which consists of an encoder that maps the normalized image to a latent representation and a prediction head that reconstructs the reconstructed image from the latent representation. The illustration is presented in Fig. 10.

Given an input image, it is divided into regular and non-overlapping patches. A subset of patches is selected, while the remaining ones are masked. The encoder embeds the visible patches using a linear projection with added positional embeddings and processes them through a series of Transformer blocks. It is noted that the encoder operates exclusively on visible, unmasked patches, as masked patches are removed, and no mask tokens are used. The encoder extracts a latent feature representation of the masked image, which is utilized to predict the original signals in the masked regions. For the encoder, we consider two common vision Transformer architectures: the standard Vision Transformer (ViT, Dosovitskiy et al., 2020) and the Swin Transformer (SwinT, Liu et al., 2021).

The prediction head processes the latent feature representation to generate a form of the original signals for the masked regions. While the prediction head can have arbitrary form and capacity, we employ a single-layer 1×1 convolutional layer to maintain a small model size. Each output element from the prediction head is a vector of pixel values corresponding to a patch. The final layer of the decoder is a linear projection with the number of output channels equal to the pixel count in a patch. The output of the prediction head is then reshaped to reconstruct the image.

The Mask Autoencoder (MAE, He et al., 2021) is another state-of-the-art model of masked image modelling, which takes a complete ViT architecture for both the encoder and prediction head. MAE demonstrates that random sampling with a high masking ratio significantly reduces redundancy, creating a task that cannot be easily solved by extrapolation from visible neighbouring patches. Accordingly, our reconstructor adopts a strategy of random masking with a 75% masking ratio, meaning 75% of the input image patches are masked, leaving only 25% visible for the model.

3.2.1. Training details

During training, we fine-tune the pre-trained models on complete apple instances to save training time.

Our loss function calculates the mean squared error (MSE) between the reconstructed and original images by measuring the average squared difference between their pixel values. It is defined as in Eq. (2).

$$\mathrm{MSE}(\mathbf{x}, \mathbf{y}) = \frac{1}{\Omega(\mathbf{x}_M)} \|\mathbf{y}_M - \mathbf{x}_M\|_2^2 \tag{2}$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{3\times H\times W}$ are the original RGB values and the predicted values, respectively; M indicates the set of masked pixels; $\Omega(\cdot)$ is the number of elements.

3.2.2. Evaluation details

During the evaluation, we introduce another two metrics to evaluate the reconstruction quality in de-normalized colour value.

- Peak-Signal-to-Noise Ratio (PSNR)
  PSNR (Hore and Ziou, 2010) is a widely used metric for evaluating the quality of image reconstruction in computer vision. It measures the similarity between the original and reconstructed images by comparing the ratio of peak signal to noise on a logarithmic scale. PSNR is defined as in Eq. (3), where 255 is the maximum pixel value for 8-bit images. A higher PSNR indicates that the reconstructed image is closer to the original, indicating better quality. Conversely, a lower PSNR indicates greater numerical differences between the images, reflecting poorer quality.

$$\mathrm{PSNR}(\mathbf{x},\mathbf{y}) = 10 \cdot \log_{10}\left(\frac{255^2}{\mathrm{MSE}(\mathbf{x},\mathbf{y})}\right) \tag{3}$$

- Structural Similarity Index Measure (SSIM)
  SSIM (Wang et al., 2004) is another well-known metric used to measure the structural similarity between the original and reconstructed images. It focuses on comparing structural information in images including luminance, contrast, and texture, which aligns more closely with human visual perception. The definition of SSIM is given in Eq. (4).

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1)(2\sigma_{\mathbf{xy}} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2)} \tag{4}$$

  where $\mu_{\mathbf{x}}$, $\mu_{\mathbf{y}}$ and $\sigma_{\mathbf{x}}^2$, $\sigma_{\mathbf{y}}^2$ are the average luminance and variance of the original and reconstructed images. $\sigma_{\mathbf{xy}}$ is the covariance between two images. C_1 and C_2 are small constants to avoid a zero denominator. The SSIM value ranges from $[-1, 1]$, and a higher value represents a more accurate replication of the original image.

3.3. Extractor

The feature extractor is implemented in a self-supervised learning framework, using the online-clustering method SwAV (Caron et al., 2021a). This method employs two parallel branches to facilitate feature learning. Specifically, the feature extractor is designed to identify representations associated with apple ripeness. The goal is to find a feature space in which fully unripe apples are positioned farthest from fully ripe apples, while ensuring that a random given apple image and its augmented variants are mapped to closely aligned locations. An overview of this process is presented in Fig. 10.

The input image is transformed into multiple augmented views $\mathbf{x}_{nt}$ (e.g., x_1 and x_2 in the figure) using transformations t sampled from a set $\mathcal{T}$ of image augmentation techniques.

These augmented views $\mathbf{x}_{nt}$ are then passed through an encoder f_θ, which consists of two standard convolutional layers, to generate nonlinear feature representations $\mathbf{z}_{nt}$ (e.g., z_1 and z_2). Then the feature representations are normalized using ℓ_2 normalization and projected onto the unit sphere.

Next, a code $\mathbf{q}_{nt}$ (e.g., q_1 and q_2) is computed by mapping the feature $\mathbf{z}_{nt}$ to a set of prototypes $\mathbf{C}$. The prototype $\mathbf{C}$ consists a set of K trainable vectors, denoted as $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$. In this work, $\mathbf{C}$ is represented as a matrix whose columns correspond to the prototype vectors $\mathbf{c}_1, \ldots, \mathbf{c}_K$. These prototypes are treated as model parameters and are updated iteratively during the training process.

In detail, a code is computed for one augmented version of an image and predicted from other augmented versions of the same image. Given two feature vectors, $\mathbf{z}_t$ and $\mathbf{z}_s$, derived from different augmentations of the same image, their corresponding codes $\mathbf{q}_t$ and $\mathbf{q}_s$ are obtained by matching these features to a set of K prototype vectors, $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$. The computation involves multiplying the feature vector $\mathbf{z}_{nt}$ with the prototype matrix $\mathbf{C}$, followed by applying the Sinkhorn–Knopp algorithm to normalize the result and produce the code $\mathbf{q}_{nt}$.

The prototype vectors represent the clustering centres of the apple images. As this method is an online method, the codes are updated only based on the image features within the current batch, distinguishing this method from offline clustering approaches that require the entire dataset to compute the codes. The loss function is defined in Eq. (5).

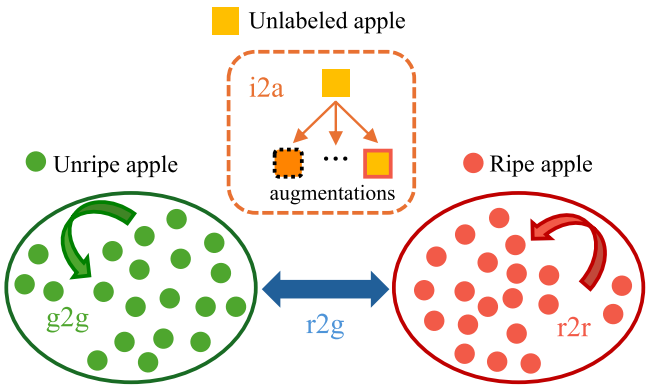


Fig. 11. The proposed distances for model performance evaluation.

$$L(\mathbf{z}_t, \mathbf{z}_s) = \ell(\mathbf{z}_t, \mathbf{q}_s) + \ell(\mathbf{z}_s, \mathbf{q}_t) \tag{5}$$

where the function $\ell(\mathbf{z}, \mathbf{q})$ quantifies the alignment between features $\mathbf{z}$ and a code $\mathbf{q}$. Conceptually, our method evaluates the similarity between the features $\mathbf{z}_t$ and $\mathbf{z}_s$ using the intermediate codes $\mathbf{q}_t$ and $\mathbf{q}_s$. In other words, if these two features are from augmentations of the same input image, and they encode the same or similar information, then it should be feasible to predict the code from the other feature.

The loss function in Eq. (5) consists of two terms that define the "swapped" prediction task: predicting the code $\mathbf{q}_t$ from the feature $\mathbf{z}_s$, and vice versa, predicting $\mathbf{q}_s$ from $\mathbf{z}_t$. Each term corresponds to the cross entropy loss between the predicted code and the probability distribution obtained by applying softmax function to the dot products of $\mathbf{z}_t$ and all prototypes in $\mathbf{C}$. The loss formulation is detailed in Eq. (6), where τ is a temperature parameter that controls the sharpness of the softmax distribution.

$$\ell(\mathbf{z}_t, \mathbf{q}_s) = -\sum_k \mathbf{q}_s^{(k)} \log \mathbf{p}_t^{(k)}, \quad \mathbf{p}_t^{(k)} = \frac{\exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_{k'}\right)} \tag{6}$$

In contrast to previous self-supervised learning methods, which directly compare the similarity of feature vectors $\mathbf{z}_{nt}$. Comparing high-dimension features (e.g. 2048) usually takes a lot of time and computational overhead. Instead, we focus on comparing the codes $\mathbf{q}_{nt}$ derived from different views, aiming to make them consistent. This strategy allows the model to capture more details of the input. In this work, we choose the code as the output of the feature extractor, as it provides a more efficient and effective representation for comparison.

Inspired by SMoG (Pang et al., 2022), the similarity comparison can happen at the instance-level, and also at the group-level. Building on this idea, we conceptualized two distances as metrics to make the extractor more suitable for our apple ripeness determination task. We illustrate the considered distances in Fig. 11.

$$D = A \sum_{i=1}^{N_G} \sum_{j \geq k}^{N_E} \left(1 - \frac{f_i \cdot f_j}{\|f_i\|_2 \|f_j\|_2}\right) \tag{7}$$

The definition of the distance is given in Eq. (7), where f is the extracted feature and $\|f\|_2$ is the Euclidean norm of f, A is a constant. k is a variable and N denotes an ordered set. Specifically, we define N_G as the set of labelled fully unripe apples and N_E as the set of labelled fully ripe apples. The sizes of N_G and N_E are 20 in our work.

- Intra-distance
  For a random unlabelled apple, we assume that the distance between the image and its augmentations should be as small as possible. This ensures that the image and its augmentations

occupy a stable position between unripe and ripe items in the feature space. This intra-distance, denoted as D_{i2a}, is implicitly considered by the loss Eq. (6).

For the set of labelled apples, we assume that unripe items should be closest to other unripe items, and ripe items should be closest to other ripe items in the feature space. To quantify this, we define that:

- The average distance between labelled unripe apples as $D = D_{g2g}$, where $i, j \in N_G, A = \frac{2}{|N_G|(|N_G|+1)}, k = i$.
- The average distance between labelled ripe apples as $D = D_{r2r}$, where $i, j \in N_G, A = \frac{2}{|N_E|(|N_E|+1)}, k = i$.

- Inter-distance

  For labelled unripe and ripe apple images, unripe items should be as distant as possible from ripe items in the feature space. To quantify this separation, we compute the average group-level distance between labelled unripe and labelled ripe apples, denoted as: $D = D_{r2g}$, where $i \in N_G, j \in N_E, A = \frac{1}{|N_G||N_E|}, k = 1$.

Intra-distances evaluate the clustering consistency within each apple and its variants in the feature space. Inter-distance measures the degree of separation between the two labelled groups, while also providing insight into the depth of the feature space. The computation of these two distances serves as a complement to the "swapped" prediction loss, offering additional metrics for assessing the effectiveness of the learned representations. This combination is also particularly useful for comparing the performance of different self-supervised learning methods.

3.3.1. Training details

In this paper, the two views consist of a global view (high-resolution, 224 × 224 pixels) and a local view (low-resolution, 112 × 112 pixels) augmentations. The extractors are trained from scratch on the set of complete apples. The backbone of the extractor is ResNet-18 to save the model size. The dimension of the output feature is set to 256, the number of prototypes is 512, the temperature τ is set to 0.1, and the number of Sinkhorn–Knopp iteration is set to 3. No pre-trained weights are used and the parameters of all convolution layers are initialized by a normal distribution.

3.3.2. Evaluation details

During the evaluation, we report the D_{r2r}, D_{g2g} and D_{r2g}, each bounded within the range [0, 2]. Ideally, lower values of D_{r2r} and D_{g2g} indicate promising performance, as they reflect the extractor's ability to effectively process the labelled images under various augmentations. Besides, D_{r2g} is expected to be significantly greater than D_{r2r} and D_{g2g}, indicating that unripe apples from ripe apples are successfully separated in the feature space. To help better compare the results, we simply compute the distance difference, defined as $(D_{r2g} - D_{r2r} - D_{g2g})$, where higher values indicate better overall separation.

These three distances serve as metrics to evaluate how closely the extractor aligns with our aim outlined in 3.1. Specifically, extractors generate high-dimension features instead of final outputs. If the extractor has lower D_{r2r}, D_{g2g} values and higher D_{r2g} value, then it is promising but does not promise to produce better final results. Because high-dimension features are then processed by the predictor for final results, the design of the predictor is also a big factor that influences final results.

3.4. Predictor

A simple 3-layer MLP predictor is employed to predict the ripeness score from the extracted features. The network consists of three fully connected layers, with dimensions set to [N, 128, 100, 1], where N represents the feature dimension from the extractor. Each layer, except the final one, is followed by a ReLU activation function. The final layer is a fully connected output layer with a single neuron, which produces the ripeness score R. This score is then normalized to fall within the range [0.0, 1.0]. This architecture effectively reduces the dimensionality from input space to a single scalar value while leveraging ReLU non-linearity to capture complex relationships between the features, ensuring robust and accurate predictions. The illustration is shown in Fig. 10.

3.4.1. Training details

During training, the weights of feature extractors are frozen, and only the weights of the predictor are updated. The predictor is trained from scratch using the labelled images only. The loss function calculates the MSE between the one-hot encoded predictions and the ground truths from labelled images.

3.4.2. Evaluation details

The mean values $\bar{x}_{green}$ and $\bar{x}_{red}$, along with the variances s^2_{green} and s^2_{red} of labelled fully unripe and fully ripe apples are selected as our evaluation metrics. Ideally, the model is expected to predict a score of 0.0 for fully unripe apples and 1.0 for fully ripe apples. These metrics align with human sense, where higher values correspond to riper apples.

The range of prediction values indicates that we treat apple ripeness prediction as a regression task rather than a multi-class classification task. As a result, our method generates continuous predictions instead of discrete ones. It avoids the inherent discontinuities of discrete classification and allows for a smooth representation of the apple ripeness distribution, providing a more nuanced understanding of ripeness levels.

Additionally and importantly, we plot the distribution of ripeness score predictions along with dense apple images, and visualize these predictions on the extracted features for better interpretation.

4. Experiments and results

4.1. Experiments

In this paper, experiments were conducted based on PyTorch Lightning 2.0.0 (Falcon, 2019) and were carried out using Python 3.9.13 and PyTorch 1.13 on a computer with an Intel Xeon Gold 6152 @2.1 GHz CPU, 1 Nvidia Tesla P100 GPU and 32.0 GB memory.

We employed a stochastic gradient descent (SGD) optimizer with a weight decay of 5×10^{-5}, a momentum of 0.9. Different initial learning rates ranging from 0.0001 to 0.06 were explored across different models to identify the optimal value for achieving the best performance.

The default patience setting for the reconstructor and extractor was set to 30 epochs to optimize training time, meaning the model terminated training if no improvement in metrics was observed after 30 epochs. In contrast, the patience for the predictor was set to 3 epochs to minimize the risk of overfitting.

4.2. Reconstructor

4.2.1. Comparison

We compare our reconstructor with MAE. The numerical results and visual comparison are shown in Table 1 and Fig. 12.

For MAE models, ViT-Base (ViT-B) with a mask size of 16 achieves a PSNR of 25.14 and a SSIM of 0.73. When the model size increases to ViT-Large (ViT-L), the performance improves, with ViT-L achieving the highest PSNR of 25.71 and an SSIM of 0.74. However, this improvement comes at the cost of significantly larger parameters, increasing from 111M to 329M.

For SimMIM models, our reconstructor with ViT-B achieves the highest SSIM of 0.75 and the second-highest PSNR of 25.36, while

Table 1
The results of reconstruction.

	Model	Image/Mask Size	PSNR↑ (dB)	SSIM↑	Params (M)
MAE	ViT-B	224/16	25.14	0.73	111
		224/32	22.00	0.67	
	ViT-L	224/16	**25.71**	0.74	329
		224/32	21.24	0.67	
SimMIM	SwinT	192[a]/16	24.40	0.74	89.9
		192[a]/32	21.47	0.72	
	ViT-B (Ours)	**224/16**	**25.36**	**0.75**	**86.3**
		224/32	21.27	0.69	

[a] Follows the pre-trained SwinT setting with a window size of 6.

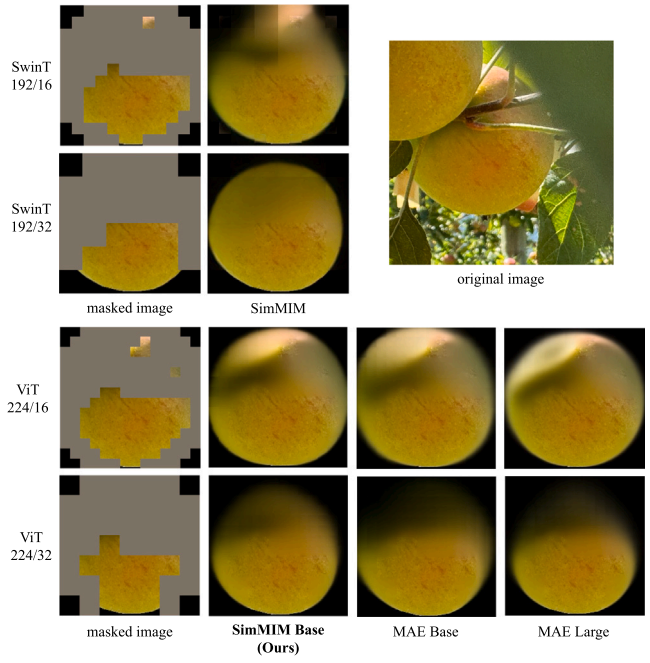


Fig. 12. The reconstruction comparison using different models and mask sizes.

utilizing only 86.3M parameters. Notably, our model has the smallest parameter count, requiring less than one-third of the parameters of MAE ViT-L, but delivering very comparable performance.

From the visual comparison, we observe that with the same input image and masking strategy, ViT-L produces the best reconstructions, while ViT-B delivers similar but reasonable results.

It is well-known that larger models deliver better performance, as they can learn and store more information. However, the small performance difference observed here is acceptable when considering the significant disparity in model size. Increasing the model size excessively for tiny marginal performance gains is not a practical choice for our task.

Compared to the standard ViT, using the Swin Transformer as the backbone yields inferior performance in our task. We hypothesize that this is due to the hierarchical structure of the Swin Transformer, which processes image patches locally using smaller patches and gradually expands the receptive field. This local processing may disrupt the consistency of information within the expanded receptive field, as illustrated in the first row of Fig. 12.

The results highlight the significant impact of mask size on performance, with larger mask sizes consistently leading to degradation across all models. The original SimMIM identifies a mask size of 32 as optimal, but based on our experiments, the performance drops substantially with a mask size of 32 compared to 16. A mask size of 16 proves to be the most suitable for reconstructing missing apple parts. We suggest that a mask size of 32 lacks flexibility, as it is too large to effectively cover the missing patches and introduces excessive noise into the visible patches.

Overall, our reconstructor achieves a favourable balance between performance and efficiency, providing valuable information for subsequent ripeness prediction.

4.2.2. Visualization

Then, we test the reconstructor with incomplete apple images under different settings. The visualization is shown in Fig. 13. Ground truths of these input images are unknown, but the detailed progress for each reconstruction is shown in the visualizations.

The various cases show diverse environmental and lighting conditions affecting the visibility and appearance of apples:

- Very limited visibility
  In cases (a) and (c), the majority of the apples are obscured, resulting in visible rates of less than 30%.
- Different occlusion sources
  In case (g), the apple is hidden by the trunk, while apples in other cases are covered by leaves.
- Lighting conditions
  In cases (a), (c), (i), and (j), the apples are shaded from direct sunlight, while in cases (d), (e), (k), and (l), they are exposed to direct sunlight.
- Shadows and light patterns
  In cases (e), (g), (h), and (k), direct shadows, light-stripes or light-spot are observed on the apples, creating complex light patterns.
- High contrast conditions
  In cases (e), (k), and (l), the apples exhibit strong contrasts between light and shadow, presenting challenging illumination scenarios.
- Backlighting effects
  In cases (b) and (g), the apples are positioned against the light source, resulting in unique lighting angles and potential silhouette effects.
- Uniform colour
  In cases (a), (c), (d), and (f), the apples are predominantly of a single colour.
- Gradual colour transitions
  In cases (b), (e), and (h), the apples showcase significant continuous colour variations, introducing additional complexity in visual features.

Our reconstructor demonstrates its reliable ability to effectively predict occluded apple parts under various conditions, including different illumination levels, occlusions, and ripeness stages in the above cases.

It is suggested that the model trained on a diverse set of apple images in various settings is able to accurately predict the occluded parts of incomplete apples. This enables the use of the trained model to reconstruct missing parts without the need for manually designed fruit shapes or handcrafted features.

4.3. Extractor

The extractor serves as a critical component of our method, acting as a bridge between the input images and the predictor. To evaluate the performance, we compare our extractor against 15 other self-supervised methods and a supervised binary classification model. For the binary classification model, we employ MSE as the loss function, while the self-supervised methods utilize their respective original loss functions, including negative cosine similarity loss, normalized temperature-scaled cross-entropy loss (NT-Xent loss), and other customized loss functions. The comparative results are presented in Table 2.

Fig. 13. The visualization of reconstruction, the numbers in masked input indicate visible rates for the model. Detailed analysis of (a)~(l) are in 4.2.2.

Table 2
The results of extractor.

Extractor	Backbone	Loss	Dimension	$D_{r2r}\downarrow$	$D_{g2g}\downarrow$	$D_{r2g}\uparrow$	$(D_{r2g}-D_{r2r}-D_{g2g})\uparrow$	Params (M)
Binary[a]	Res18	MSE loss	512	0.1488	0.0609	0.2391	0.0293	**11.2**
BYOL (Grill et al., 2020)	Res18	NegativeCosineSimilarity	256	0.0002	0.0036	0.0038	−0.0001	12.5
FastSiam (Pototzky et al., 2022)	Res18	NegativeCosineSimilarity	128	0.0504	0.0032	0.1370	0.0833	11.8
SimSiam (Chen and He, 2021)	Res18	NegativeCosineSimilarity	256	0.0118	0.1594	0.7084	**0.5373**	13.5
DenseCL (Wang et al., 2021)	Res18	NT-Xent loss	2048	0.2488	0.2035	0.6969	0.2446	23.7
MoCo (He et al., 2020)	Res18	NT-Xent loss	512	0.5583	0.8731	0.9668	−0.4646	11.5
NNCLR (Dwibedi et al., 2021)	Res18	NT-Xent loss	128	0.3879	0.8371	**1.1086**	−0.1163	11.9
SimCLR (Chen et al., 2020)	Res18	NT-Xent loss	512	0.2769	0.2366	0.6583	0.1448	11.5
DCL (Yeh et al., 2022)	Res18	DCL loss	512	0.8192	1.0193	0.7207	−1.1179	11.5
DCLW (Yeh et al., 2022)	Res18	DCL weighted loss	512	1.0032	1.4614	0.8880	−1.5767	11.5
DINO (Caron et al., 2021b)	Res18	DINO loss	2048	0.5046	0.2689	1.1046	0.3311	23.7
MSN (Assran et al., 2022b)	ViT-S	MSN loss	256	0.6997	1.1675	1.0978	−0.7694	27.8
PMSN (Assran et al., 2022a)	ViT-S	PMSN loss	384	**0.0001**	**0.0003**	0.0001	−0.0003	27.8
TiCo (Zhu et al., 2022)	Res18	TiCo loss	256	0.4499	0.3836	0.5986	−0.2349	23.9
VICReg (Bardes et al., 2022a)	Res18	VICReg loss	512	0.1828	0.1722	0.1849	−0.1701	16.4
VICRegL (Bardes et al., 2022b)	Res18	VICRegL loss	2048	0.6578	0.6641	0.6012	−0.7208	20.7
SwAV(Ours)	**Res18**	**SwAV loss**	**256**	**0.3816**	**0.2418**	**0.8844**	**0.2610**	**11.7**

[a] Binary classification is the only supervised model, using features extracted from the layer before fully connected layer.

Table 3
The results of predictor using features from extractors.

Extractor	$\bar{x}_{green}\downarrow$	$s^2_{green}\downarrow$	$\bar{x}_{red}\uparrow$	$s^2_{red}\downarrow$
Binary	0.2460	0.0037	0.7258	0.0098
BYOL	0.0636	0.0017	0.6383	0.0414
FastSiam	0.5336	0.0014	0.8011	0.0052
SimSiam	0.2375	0.0011	0.7302	0.0047
DenseCL	0.3329	0.0031	0.7440	0.0185
MoCo	0.1964	0.0055	0.6536	0.0145
NNCLR	0.1194	0.0012	0.7821	0.0162
SimCLR	0.0607	0.0012	0.7548	0.0169
DINO	0.1798	0.0037	0.8208	0.0161
TiCo	0.0444	0.0005	0.7606	**0.0034**
VICReg	0.0893	0.0011	0.7121	0.0070
SwAV(Ours)	**0.0127**	**0.0001**	**0.8933**	**0.0094**

ResNet-18 is selected as the backbone for most of the self-supervised methods, as it is more lightweight compared to the commonly used ResNet-50. For MSN and PMSN, ViT-Small (ViT-S) is used, following their respective model designs. The output dimensions for each method are kept consistent with their original configurations.

The results demonstrate that supervised binary classification and several self-supervised methods demonstrate strong performance in separating fully unripe and fully ripe apples within the feature space. However, certain self-supervised methods, such as DCL, DCLW, MSN, PMSN, and VICRegL, fail to meet expectations for this task. Their D_{r2g} values are smaller than D_{r2r} and D_{g2g}, indicating an insufficient separation between unripe and ripe apples, thus they are excluded to be incorporated with the predictor.

The binary classification model achieves the D_{r2g} (0.2391) greater than both D_{r2r} (0.1488) and D_{g2g} (0.0609). These results suggest that unripe apples are distributed more densely than ripe apples. Our method achieves the D_{r2g} of 0.8844, which is significantly greater

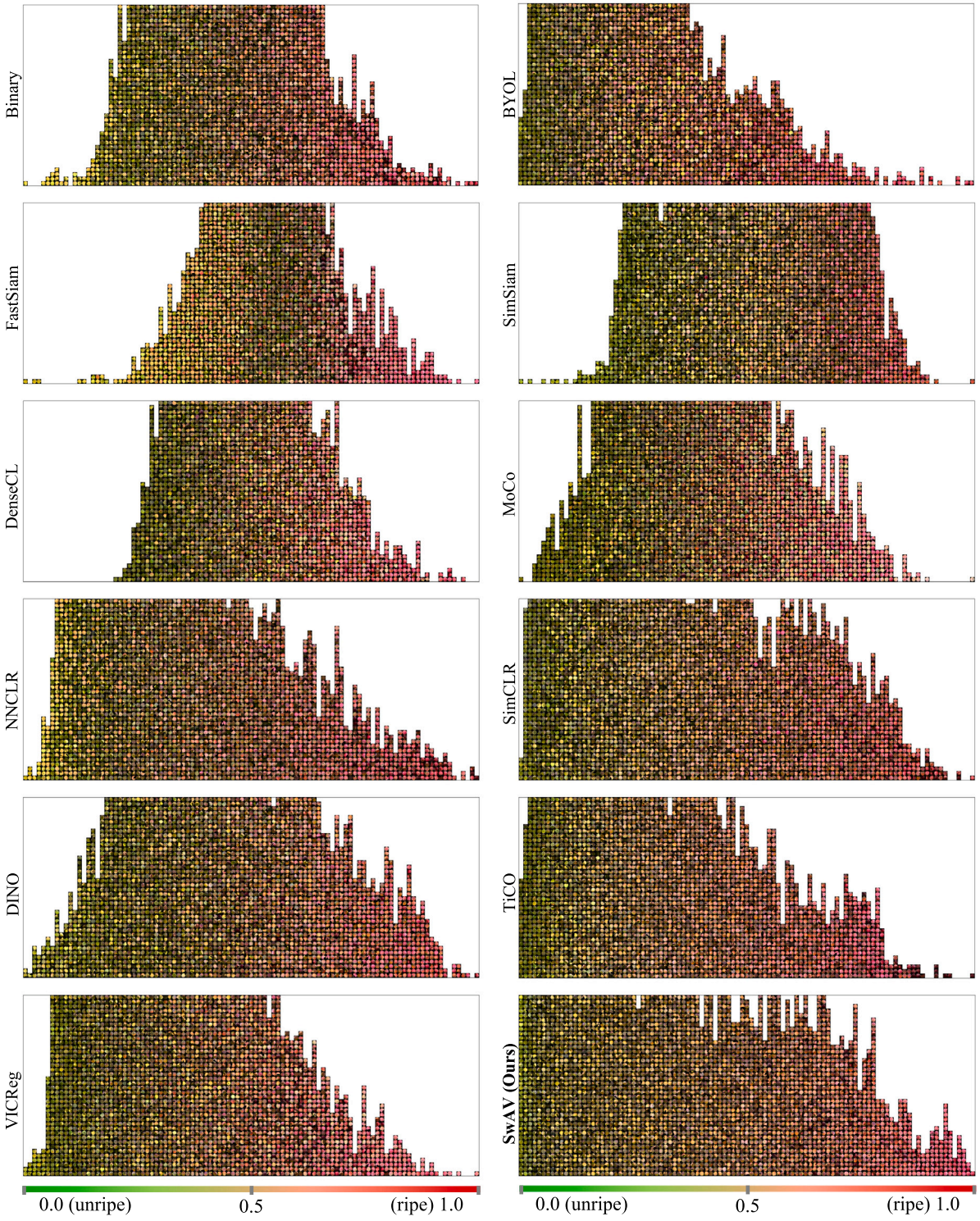


Fig. 14. Ripeness score R predictions for complete apple instances, with intervals of 0.1 and at most 40 items displayed per score.

than D_{g2g} (0.3816) and D_{r2r} (0.2418), demonstrating a better balance between the clustering of ripe and unripe apples compared to binary classification.

While PMSN achieves the smallest D_{g2g} and D_{r2r}, its D_{r2g} equals D_{r2r}, indicating that it does not effectively separate unripe and ripe apples in the feature space. NNCLR achieves the highest D_{r2g} of 1.1086, but the margin relative to its D_{r2r} and D_{g2g} is insufficient to ensure clear separation.

SimSiam achieves the highest distance difference of 0.5373, with a remarkably low D_{r2r} of 0.0118. It is noted that DINO also demonstrates a balanced distribution between unripe and ripe apples, reflecting its ability to achieve meaningful separation. In contrast, the binary classification method yields a distance difference of only 0.0293 due to

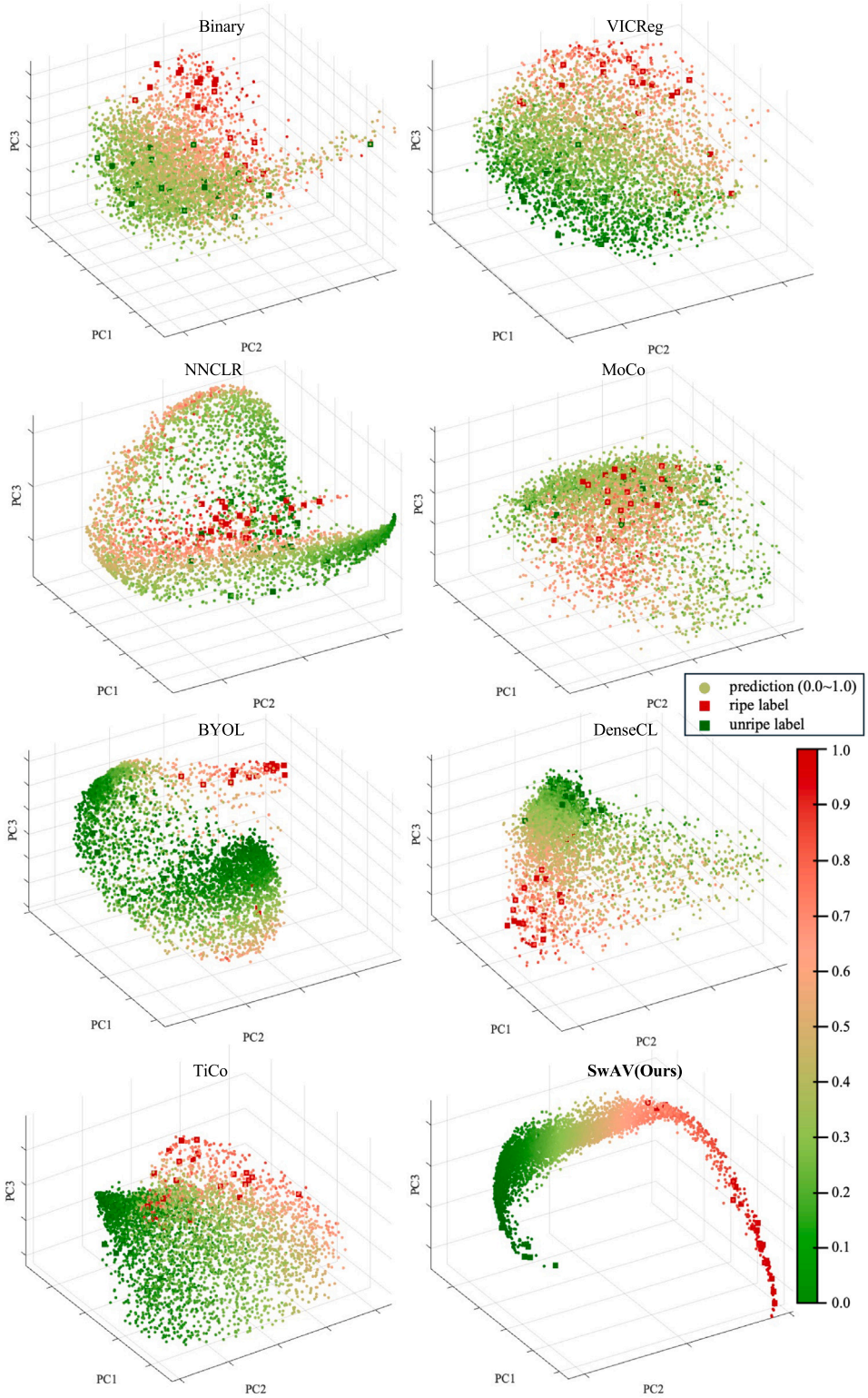


Fig. 15. 3D PCA visualizations of ripeness scores on extracted features.

imbalanced D_{g2g} and D_{r2r}. Our method achieves a distance difference of 0.2610, significantly outperforming the binary classification approach by a large margin. It also surpasses several other self-supervised methods, showcasing robust performance in separating unripe and ripe apples.

Regarding model size, introducing complex backbones, such as ViT-S with 27.8M parameters, does not bring noticeable improvements. We suggest that this is because our task is relatively simple, making heavy backbones prone to over-fitting. Additionally, The binary classification model only occupies 11.2M parameters as the result of no extra modules being introduced. Our method is with 11.7M parameters, incorporating additional parameters for the extra branch and prototypes **C**. Despite this, our model remains more compact than many other self-supervised methods while delivering superior performance.

4.4. Predictor

4.4.1. Comparison

The 12 extractors with $D_{r2g} > D_{r2r}$ and $D_{r2g} > D_{g2g}$ are selected to extract image features for the predictor. The performance of the predictor is summarized in Table 3. Our method demonstrates the best overall performance, achieving the lowest $\bar{x}_{green}$ of 0.0127 and s^2_{green} of 0.0001, along with the highest $\bar{x}_{red}$ of 0.8933 and the second-highest s^2_{red} of 0.0094. In contrast, the binary classification model yields a $\bar{x}_{green}$ of 0.2460 and a $\bar{x}_{red}$ of 0.7258, indicating its comparatively weaker capability in predicting ripeness scores.

The results further highlight that some self-supervised methods outperform the binary classification model. For example, TiCO achieves competitive results with the lowest s^2_{red} of 0.0034 and the second-lowest $\bar{x}_{green}$ of 0.0127. DINO delivers a $\bar{x}_{green}$ of 0.1798 and a x_{red} of 0.8208. Similarly, VICReg and SimCLR produce relatively low $\bar{x}_{green}$ values and high $\bar{x}_{red}$ values.

4.4.2. Visualization

To present the results more clearly, the ripeness score predictions are visualized in Fig. 14.

The analysis of these predictions is conducted from the following three perspectives:

- Prediction continuity
  The dataset contains apples at various ripeness stages, with 40 labelled fully unripe and fully ripe apples used for training. Consequently, the predictions are expected to span the entire range of scores, from 0.0 (unripe) to 1.0 (ripe), reflecting a continuous progression.
  Among the evaluated methods, our approach uniquely achieves seamless and continuous predictions across the entire score range, accurately representing all ripeness stages. Other methods, including NNCLR, DINO, SimCLR, VICReg, and the binary classification model, also approximate full-score predictions but exhibit gaps, with certain score intervals missing in their outputs. This discontinuity indicates limitations in capturing the smooth progression of ripeness.
- Prediction distribution
  Like many large image datasets, including our previous Nine-Peach dataset (Zhao et al., 2023b), the apple dataset should exhibit a "long-tail" distribution. This reflects the natural tendency for unripe apples to outnumber ripe ones due to factors such as natural fruit-falling and artificial fruit-thinning.
  Several methods, including binary classification, FastSiam, SimSiam, DenseCL, MoCo, and DINO, produce predictions with a Gaussian-like distribution. These methods do not generate sufficient predictions for unripe apples. Most predictions fall in the semi-ripe range, indicating poor separation between unripe and ripe apples. In contrast, our method, along with BYOL, NNCLR, SimCLR, TiCO, and VICReg, predicts ripeness scores following the expected "long-tail" distribution. The predicted number of apples gradually decreases from unripe to ripe, effectively reflecting the natural progression of apple ripening.
- Colour gradient
  A smooth colour gradient from unripe to ripe is an essential indicator of the accuracy of ripeness predictions. Ideally, the gradient should transition smoothly from green for unripe apples to red for fully ripe ones.
  Some methods, including BYOL, FastSiam, MoCo, and VICReg, exhibit obvious inconsistencies, as some green apples are incorrectly assigned scores over 0.5, suggesting outliers of prediction. SimCLR and TiCO also face challenges, with semi-ripe and ripe apples often mixed, making it difficult to tell. Notably, our method delivers a smooth and consistent colour gradient. The predictions start with green on the left and gradually transition to red on the right, accurately reflecting the natural ripening process. This demonstrates the robustness and precision of our approach in ripeness estimation.

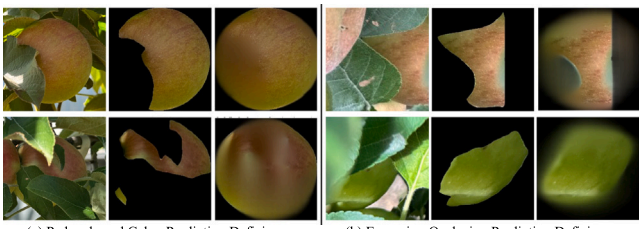


Fig. 16. Two prediction deficiencies in our reconstructor.

We used 3D Principal Component Analysis (PCA) to reduce the dimensionality of the extracted features to three dimensions, with the visualization presented in Fig. 15.

Among all of the visualizations, our method stands out by generating a smooth manifold where apple ripeness increases progressively. In the space, the labelled unripe and ripe apples are distinctly separated, indicating high explainability for the ripeness score predictions.

Since ripeness score prediction is a subjective topic, we invited several volunteers including apple-picking robot professionals and normal apple consumers to choose the best prediction from their perspectives. The test was conducted anonymously, and ground truths were not disclosed. All of the participants agreed that our predictor and TiCO are the top-performing methods. However, compared to our predictor, although TiCO shows a good colour gradient, it is unconfident with accurate predictions for ripe apples, as a result of $\bar{x}_{red}$ of 0.7606.

The results further highlight that self-supervised methods can outperform supervised binary classification. This underscores the ability of self-supervised models to learn latent ripeness-related features from a large number of unlabelled images, significantly reducing the need for manual labelling.

5. Discussion

5.1. Limitation

Our apple images were collected from a single Jazz apple orchard, which may not represent the different varieties of apples. Despite extensive searches, we could not find public datasets that meet our research requirements. This constraint has led us to rely only on our collected dataset.

In terms of reconstruction, there are two prediction deficiencies as shown in Fig. 16. The first is peduncle and calyx prediction deficiency, the model cannot predict the apple peduncle and calyx as expected. The second deficiency appears when excessive occlusion occurs, with very

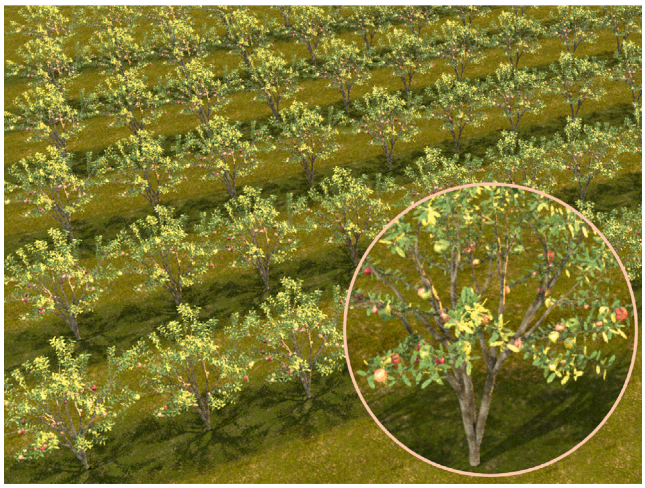


Fig. 17. The digital simulation of a large orchard, with apple locations and ripeness monitored.

limited visible information, the reconstructor cannot perform well and generate reasonable results.

Our method is designed for in-field apples which have significant colour changes during their ripening progress. Therefore, it is not suitable for certain apple cultivars like Granny Smith, which remain green throughout all ripening stages. Additionally, it cannot be applied to fruits that ripen after harvesting like bananas, or to those evaluated based on softness like avocados.

5.2. Future work

To improve our method's applicability, we will expand the dataset by including a more diverse range of apple varieties, capturing a broader representation across different types.

Our method demonstrates that it is feasible to use a single-view image to predict apple ripeness. We propose the next work should focus on extending this method to work with multi-view images, which would allow more accurate ripeness estimation. This method has the potential to be extended to other fruits that exhibit significant colour changes during the ripening process, such as peaches.

Besides, the proposed method is promising for deployment on in-field robots to capture both the ripeness and spatial information of apples, making it possible to monitor the ripeness distribution across a large orchard. This information can facilitate data-driven decision-making for orchard management and then be used to guide autonomous picking-robots to selectively harvest ripe apples. We simulate such kind of apple orchard in a 3D digital environment, as shown in Fig. 17.

6. Conclusion

Developing apple-harvesting robots capable of identifying the ripeness stage of apples is a challenging task, particularly because in-field apples are often obscured by leaves, branches, or trunks. Determining apple ripeness is also challenging as it is subjective to define the number of ripeness stages. Under this context, we propose a novel self-supervised method utilizing 40 labelled and 7151 unlabelled apple images for two problems: ripeness determination and in-field occlusion.

Our method consists of three key parts: a reconstructor, a feature extractor, and a predictor. The reconstructor is trained to restore the missing details of occluded apples, enabling more complete visual representations. The feature extractor leverages a vast number of unlabelled images to learn ripeness-related features effectively, reducing the reliance on labelled images. Finally, the predictor uses the extracted features to generate flexible ripeness scores between 0.0 and 1.0, eliminating the need for subjectively pre-defined ripeness stages. This flexibility allows end-users to make customized decisions according to their specific needs and criteria.

Experimental results highlight that our method achieves the highest SSIM of 0.75 and the second-highest PSNR of 25.36 for reconstructing incomplete apples, with the fewest 86.3M parameters. Besides, our method outperforms 15 other self-supervised methods and even a supervised method in ripeness score prediction, achieving the lowest score of 0.0127 for fully unripe apples and the highest score of 0.8933 for fully ripe apples.

Our method is promising for integration into in-field robotic systems, enabling them to determine ripeness effectively and selectively harvest only ripe fruits. Furthermore, it can be used to monitor overall ripeness trends across large orchards, helping managers make informed decisions about harvest timing and orchard management. Our method contributes to the goals of smart precision agriculture.

CRediT authorship contribution statement

Ziang Zhao: Writing – original draft, Visualization, Validation, Software, Methodology, Data curation, Conceptualization. **Yulia Hicks:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **Xianfang Sun:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Benjamin J. McGuinness:** Writing – review & editing, Resources, Methodology, Data curation, Conceptualization. **Hin S. Lim:** Writing – review & editing, Resources, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We extend our gratitude to Rahul Jangali for his invaluable assistance with image collection, and we thank T&G Orchard for providing access to the apple trees. This research was supported by Cardiff University and University of Waikato Collaboration Seed Fund. We appreciate the computational resources provided by Advanced Research Computing at Cardiff (ARCCA).

Data availability

Data will be made available on request.

References

Assran, M., Balestriero, R., Duval, Q., Bordes, F., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., Ballas, N., 2022a. The hidden uniform cluster prior in self-supervised learning. http://dx.doi.org/10.48550/arXiv.2210.07277, http://arxiv.org/abs/2210.07277, arXiv:2210.07277.

Assran, M., Caron, M., Misra, I., Bojanowski, P., Bordes, F., Vincent, P., Joulin, A., Rabbat, M., Ballas, N., 2022b. Masked siamese networks for label-efficient learning. In: Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October (2022) 23–27, Proceedings, Part XXXI. Springer-Verlag, Berlin, Heidelberg, pp. 456–473. http://dx.doi.org/10.1007/978-3-031-19821-2_26.

Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., Schwarzschild, A., Wilson, A.G., Geiping, J., Garrido, Q., Fernandez, P., Bar, A., Pirsiavash, H., LeCun, Y., Goldblum, M., 2023. A cookbook of self-supervised learning. http://arxiv.org/abs/2304.12210. arXiv:2304.12210, [cs].

Bardes, A., Ponce, J., LeCun, Y., 2022a. VICReg: variance-invariance-covariance regularization for self-supervised learning. http://dx.doi.org/10.48550/arXiv.2105.04906, http://arxiv.org/abs/2105.04906, arXiv:2105.04906.

Bardes, A., Ponce, J., LeCun, Y., 2022b. VICRegL: self-supervised learning of local visual features. http://dx.doi.org/10.48550/arXiv.2210.01571, http://arxiv.org/abs/2210.01571, arXiv:2210.01571.

Bu, L., Chen, C., Hu, G., Sugirbay, A., Sun, H., Chen, J., 2022. Design and evaluation of a robotic apple harvester using optimized picking patterns. Comput. Electron. Agric. 198, 107092. http://dx.doi.org/10.1016/j.compag.2022.107092, https://linkinghub.elsevier.com/retrieve/pii/S0168169922004094.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A., 2021a. Unsupervised learning of visual features by contrasting cluster assignments. http://arxiv.org/abs/2006.09882. arXiv:2006.09882, [cs].

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A., 2021b. Emerging properties in self-supervised vision transformers. http://arxiv.org/abs/2104.14294. arXiv:2104.14294, [cs].

Chen, M., Chen, Z., Luo, L., Tang, Y., Cheng, J., Wei, H., Wang, J., 2024. Dynamic visual servo control methods for continuous operation of a fruit harvesting robot working throughout an orchard. Comput. Electron. Agric. 219, 108774. http://dx.doi.org/10.1016/j.compag.2024.108774, https://linkinghub.elsevier.com/retrieve/pii/S0168169924001650.

Chen, X., He, K., 2021. Exploring simple siamese representation learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Nashville, TN, USA, pp. 15745–15753. http://dx.doi.org/10.1109/CVPR46437.2021.01549, https://ieeexplore.ieee.org/document/9578004/.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020. A simple framework for contrastive learning of visual representations. http://arxiv.org/abs/2002.05709. arXiv:2002.05709, [cs, stat].

Chen, S., Zou, X., Zhou, X., Xiang, Y., Wu, M., 2023. Study on fusion clustering and improved YOLOv5 algorithm based on multiple occlusion of Camellia oleifera fruit. Comput. Electron. Agric. 206, 107706. http://dx.doi.org/10.1016/j.compag.2023.107706, https://linkinghub.elsevier.com/retrieve/pii/S0168169923000947.

Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., Shan, Y., 2024. YOLO-world: real-time open-vocabulary object detection. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Seattle, WA, USA, pp. 16901–16911. http://dx.doi.org/10.1109/CVPR52733.2024.01599, https://ieeexplore.ieee.org/document/10657649/.

Choi, T., Would, O., Salazar-Gomez, A., Liu, X., Cielniak, G., 2024. Channel randomisation: Self-supervised representation learning for reliable visual anomaly detection in speciality crops. Comput. Electron. Agric. 226, 109416. http://dx.doi.org/10.1016/j.compag.2024.109416, https://linkinghub.elsevier.com/retrieve/pii/S016816992400807X.

Das, A.J., Wahi, A., Kothari, I., Raskar, R., 2016. Ultra-portable, wireless smartphone spectrometer for rapid, non-destructive testing of fruit ripeness. Sci. Rep. 6, 32504. http://dx.doi.org/10.1038/srep32504, https://www.nature.com/articles/srep32504.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2020. An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=YicbFdNTTy.

Du, X., Cheng, H., Ma, Z., Lu, W., Wang, M., Meng, Z., Jiang, C., Hong, F., 2023. DSW-YOLO: A detection method for ground-planted strawberry fruits under different occlusion levels. Comput. Electron. Agric. 214, 108304. http://dx.doi.org/10.1016/j.compag.2023.108304, https://linkinghub.elsevier.com/retrieve/pii/S0168169923006920.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A., 2021. With a little help from my friends: nearest-neighbor contrastive learning of visual representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9588–9597, https://openaccess.thecvf.com/content/ICCV2021/html/Dwibedi_With_a_Little_Help_From_My_Friends_Nearest-Neighbor_Contrastive_Learning_ICCV_2021_paper.html.

Falcon, W., 2019. PyTorch Lightning. https://lightning.ai/.

FAO, 2024. Production/Value of Agricultural Production. https://www.fao.org/faostat/en/#data/QV.

Gai, R.L., Wei, K., Wang, P.F., 2023. SSMDA: self-supervised cherry maturity detection algorithm based on multi-feature contrastive learning. Agriculture 13, 939. http://dx.doi.org/10.3390/agriculture13050939, https://www.mdpi.com/2077-0472/13/5/939.

Gené-Mola, J., Ferrer-Ferrer, M., Gregorio, E., Blok, P.M., Hemming, J., Morros, J.R., Rosell-Polo, J.R., Vilaplana, V., Ruiz-Hidalgo, J., 2023. Looking behind occlusions: A study on amodal segmentation for robust on-tree apple fruit size estimation. Comput. Electron. Agric. 209, 107854. http://dx.doi.org/10.1016/j.compag.2023.107854, https://linkinghub.elsevier.com/retrieve/pii/S0168169923002429.

Girshick, R., 2015. Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1440–1448. http://dx.doi.org/10.1109/ICCV.2015.169, https://ieeexplore.ieee.org/document/7410526.

Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M., 2020. Bootstrap your own latent a new approach to self-supervised learning. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, pp. 21271–21284.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R., 2021. Masked autoencoders are scalable vision learners. http://arxiv.org/abs/2111.06377. arXiv:2111.06377, [cs].

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738, https://openaccess.thecvf.com/content_CVPR_2020/html/He_Momentum_Contrast_for_Unsupervised_Visual_Representation_Learning_CVPR_2020_paper.html.

Hore, A., Ziou, D., 2010. Image quality metrics: PSNR vs. SSIM. In: 2010 20th International Conference on Pattern Recognition. IEEE, Istanbul, Turkey, pp. 2366–2369. http://dx.doi.org/10.1109/ICPR.2010.579, http://ieeexplore.ieee.org/document/5596999/.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861, [cs] http://arxiv.org/abs/1704.04861.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2261–2269. http://dx.doi.org/10.1109/CVPR.2017.243, https://ieeexplore.ieee.org/document/8099726.

Jangali, R., McGuinness, B., Lim, H., Williams, H., Qureshi, A.H., Smith, D., MacDonald, B.A., Duke, M., 2024. Development of a novel multipurpose robotic end effector for fruitlet thinning and fruit harvesting of apples. In: 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE). pp. 2073–2078. http://dx.doi.org/10.1109/CASE59546.2024.10711387, https://ieeexplore.ieee.org/document/10711387.

Kang, H., Zhou, H., Wang, X., Chen, C., 2020. Real-time fruit recognition and grasping estimation for robotic apple harvesting. Sensors 20, 5670. http://dx.doi.org/10.3390/s20195670, https://www.mdpi.com/1424-8220/20/19/5670.

Kim, S., Hong, S.J., Ryu, J., Kim, E., Lee, C.H., Kim, G., 2023. Application of amodal segmentation on cucumber segmentation and occlusion recovery. Comput. Electron. Agric. 210, 107847. http://dx.doi.org/10.1016/j.compag.2023.107847, https://linkinghub.elsevier.com/retrieve/pii/S0168169923002351.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R., 2023. Segment anything. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3992–4003. http://dx.doi.org/10.1109/ICCV51070.2023.00371, https://ieeexplore.ieee.org/document/10378323.

Lammers, K., Zhang, K., Zhu, K., Chu, P., Li, Z., Lu, R., 2024. Development and evaluation of a dual-arm robotic apple harvesting system. Comput. Electron. Agric. 227, 109586. http://dx.doi.org/10.1016/j.compag.2024.109586, https://linkinghub.elsevier.com/retrieve/pii/S0168169924009773.

Li, H., Gu, Z., He, D., Wang, X., Huang, J., Mo, Y., Li, P., Huang, Z., Wu, F., 2024. A lightweight improved YOLOv5s model and its deployment for detecting pitaya fruits in daytime and nighttime light-supplement environments. Comput. Electron. Agric. 220, 108914. http://dx.doi.org/10.1016/j.compag.2024.108914, https://linkinghub.elsevier.com/retrieve/pii/S0168169924003053.

Li, K., Malik, J., 2016. Amodal instance segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp. 677–693. http://dx.doi.org/10.1007/978-3-319-46475-6_42.

Liang, J., Huang, K., Lei, H., Zhong, Z., Cai, Y., Jiao, Z., 2024. Occlusion-aware fruit segmentation in complex natural environments under shape prior. Comput. Electron. Agric. 217, 108620. http://dx.doi.org/10.1016/j.compag.2024.108620, https://linkinghub.elsevier.com/retrieve/pii/S0168169924000115.

Liu, Y., Chen, N., Ma, Z., Che, F., Mao, J., Chen, B., 2016. The changes in color, soluble sugars, organic acids, anthocyanins and aroma components in starkrimson during the ripening period in China. Molecules vol. 21, 812. http://dx.doi.org/10.3390/molecules21060812, https://www.mdpi.com/1420-3049/21/6/812.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE Computer Society, pp. 9992–10002. http://dx.doi.org/10.1109/ICCV48922.2021.00986, https://www.computer.org/csdl/proceedings-article/iccv/2021/281200j992/1BmGKZoEzug.

Liu, C., Liu, W., Chen, W., Yang, J., Zheng, L., 2015. Feasibility in multispectral imaging for predicting the content of bioactive compounds in intact tomato fruit. Food Chem. 173, 482–488. http://dx.doi.org/10.1016/j.foodchem.2014.10.052, https://linkinghub.elsevier.com/retrieve/pii/S0308814614016148.

Meng, F., Li, J., Zhang, Y., Qi, S., Tang, Y., 2023. Transforming unmanned pineapple picking with spatio-temporal convolutional neural networks. Comput. Electron. Agric. 214, 108298. http://dx.doi.org/10.1016/j.compag.2023.108298, https://linkinghub.elsevier.com/retrieve/pii/S0168169923006865.

Miraei Ashtiani, S.H., Javanmardi, S., Jahanbanifard, M., Martynenko, A., Verbeek, F.J., 2021. Detection of mulberry ripeness stages using deep learning models. IEEE Access 9, 100380–100394. http://dx.doi.org/10.1109/ACCESS.2021.3096550, conference Name: IEEE Access.

Pang, B., Zhang, Y., Li, Y., Cai, J., Lu, C., 2022. Unsupervised visual representation learning by synchronous momentum grouping. In: Computer Vision – ECCV 2022. Springer, Cham, pp. 265–282. http://dx.doi.org/10.1007/978-3-031-20056-4_16, https://link.springer.com/chapter/10.1007/978-3-031-20056-4_16.

Pototzky, D., Sultan, A., Schmidt-Thieme, L., 2022. FastSiam: Resource-efficient self-supervised learning on a single GPU. In: Pattern Recognition: 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September (2022) 27–30, Proceedings. Springer-Verlag, Berlin, Heidelberg, pp. 53–67. http://dx.doi.org/10.1007/978-3-031-16788-1_4.

Qin, J., Lu, R., Peng, Y., 2009. Prediction of apple internal quality using spectral absorption and scattering properties. Trans. ASABE 52, 486–499. http://dx.doi.org/10.13031/2013.26807, http://elibrary.asabe.org/abstract.asp??JID=3&AID=26807&CID=t2009&v=52&i=2&T=1.

Ramos, R.P., Gomes, J.S., Prates, R.M., Simas Filho, E.F., Teruel, B.J., dos Santos Costa, D., 2021. Non-invasive setup for grape maturation classification using deep learning. J. Sci. Food Agric. 101, 2042–2051. http://dx.doi.org/10.1002/jsfa.10824, https://onlinelibrary.wiley.com/doi/abs/10.1002/jsfa.10824, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jsfa.10824.

Saranya, N., Srinivasan, K., Kumar, S.K.P., 2021. Banana ripeness stage identification: a deep learning approach. J. Ambient. Intell. Humaniz. Comput. http://dx.doi.org/10.1007/s12652-021-03267-w.

Silwal, A., Davidson, J.R., Karkee, M., Mo, C., Zhang, Q., Lewis, K., 2017. Design, integration, and field evaluation of a robotic apple harvester. J. Field Robot. 34, 1140–1159. http://dx.doi.org/10.1002/rob.21715, https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21715, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21715.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. http://arxiv.org/abs/1409.1556. arXiv:1409.1556.

Suharjito, Elwirehardja, G.N., Prayoga, J.S., 2021. Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. Comput. Electron. Agric. 188, 106359. http://dx.doi.org/10.1016/j.compag.2021.106359, https://www.sciencedirect.com/science/article/pii/S0168169921003768.

Sun, T., Zhang, W., Gao, X., Zhang, W., Li, N., Miao, Z., 2024. Efficient occlusion avoidance based on active deep sensing for harvesting robots. Comput. Electron. Agric. 225, 109360. http://dx.doi.org/10.1016/j.compag.2024.109360, https://linkinghub.elsevier.com/retrieve/pii/S0168169924007518.

Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z., 2019. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput. Electron. Agric. 157, 417–426. http://dx.doi.org/10.1016/j.compag.2019.01.012, https://www.sciencedirect.com/science/article/pii/S016816991831528X.

Van Herck, L., Kurtser, P., Wittemans, L., Edan, Y., 2020. Crop design for improved robotic harvesting: A case study of sweet pepper harvesting. Biosyst. Eng. 192, 294–308. http://dx.doi.org/10.1016/j.biosystemseng.2020.01.021, https://linkinghub.elsevier.com/retrieve/pii/S1537511020300337.

Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13, 600–612. http://dx.doi.org/10.1109/TIP.2003.819861, http://ieeexplore.ieee.org/document/1284395/.

Wang, A., Qian, W., Li, A., Xu, Y., Hu, J., Xie, Y., Zhang, L., 2024. NVW-YOLOv8s: An improved YOLOv8s network for real-time detection and segmentation of tomato fruits at different ripeness stages. Comput. Electron. Agric. 219, 108833. http://dx.doi.org/10.1016/j.compag.2024.108833, https://linkinghub.elsevier.com/retrieve/pii/S0168169924002242.

Wang, D., Wang, X., Chen, Y., Wu, Y., Zhang, X., 2023. Strawberry ripeness classification method in facility environment based on red color ratio of fruit rind. Comput. Electron. Agric. 214, 108313. http://dx.doi.org/10.1016/j.compag.2023.108313, https://linkinghub.elsevier.com/retrieve/pii/S0168169923007019.

Wang, X., Zhang, R., Shen, C., Kong, T., Li, L., 2021. Dense contrastive learning for self-supervised visual pre-training. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Nashville, TN, USA, pp. 3023–3032. http://dx.doi.org/10.1109/CVPR46437.2021.00304, https://ieeexplore.ieee.org/document/9578497/.

Xiao, B., Nguyen, M., Yan, W.Q., 2021. Apple ripeness identification using deep learning. In: Geometry and Vision. Springer International Publishing, Cham, pp. 53–67. http://dx.doi.org/10.1007/978-3-030-72073-5_5.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H., 2022. SimMIM: a simple framework for masked image modeling. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9643–9653. http://dx.doi.org/10.1109/CVPR52688.2022.00943, https://ieeexplore.ieee.org/document/9880205.

Yeh, C.H., Hong, C.Y., Hsu, Y.C., Liu, T.L., Chen, Y., LeCun, Y., 2022. Decoupled contrastive learning. In: Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October (2022) 23–27, Proceedings, Part XXVI. Springer-Verlag, Berlin, Heidelberg, pp. 668–684. http://dx.doi.org/10.1007/978-3-031-19809-0_38.

Zhang, K., Lammers, K., Chu, P., Li, Z., Lu, R., 2021. System design and control of an apple harvesting robot. Mechatronics 79, 102644. http://dx.doi.org/10.1016/j.mechatronics.2021.102644, https://linkinghub.elsevier.com/retrieve/pii/S0957415821001173.

Zhao, Z., Hicks, Y., Sun, X., Luo, C., 2023b. Peach ripeness classification based on a new one-stage instance segmentation model. Comput. Electron. Agric. 214, 108369. http://dx.doi.org/10.1016/j.compag.2023.108369, https://www.sciencedirect.com/science/article/pii/S0168169923007573.

Zhao, R., Zhu, Y., Li, Y., 2023a. CLA: A self-supervised contrastive learning method for leaf disease identification with domain adaptation. Comput. Electron. Agric. 211, 107967. http://dx.doi.org/10.1016/j.compag.2023.107967, https://linkinghub.elsevier.com/retrieve/pii/S0168169923003551.

Zheng, C., Chen, P., Pang, J., Yang, X., Chen, C., Tu, S., Xue, Y., 2021. A mango picking vision algorithm on instance segmentation and key point detection from RGB images in an open orchard. Biosyst. Eng. 206, 32–54. http://dx.doi.org/10.1016/j.biosystemseng.2021.03.012, https://linkinghub.elsevier.com/retrieve/pii/S1537511021000738.

Zhu, J., Moraes, R.M., Karakulak, S., Sobol, V., Canziani, A., LeCun, Y., 2022. TiCo: Transformation invariance and covariance contrast for self-supervised visual representation learning. http://dx.doi.org/10.48550/arXiv.2206.10698, http://arxiv.org/abs/2206.10698, arXiv:2206.10698.