applied sciences

MDPI

Article

HFD: Hierarchical Feature Detector for Stem End of Pomelo with Transformers

Bowen Hou [1,2,*] and Gongyan Li [2]

[1] University of Chinese Academy of Sciences, Beijing 100049, China
[2] Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China
* Correspondence: houbowen@ime.ac.cn

Abstract: Transformers have become increasingly prevalent in computer vision research, especially for object detection. To accurately and efficiently distinguish the stem end of pomelo from its black spots, we propose a hierarchical feature detector, which reconfigures the self-attention model, with high detection accuracy. We designed the combination attention module and the hierarchical feature fusion module that utilize multi-scale features to improve detection performance. We created a dataset in COCO format and annotated two types of detection targets: the stem end and the black spot. Experimental results on our pomelo dataset confirm that HFD's results are comparable to those of state-of-the-art one-stage detectors such as YOLO v4 and YOLO v5 and transformer-based detectors such as DETR, Deformable DETR, and YOLOS. It achieves 89.65% mAP at 70.92 FPS with 100.34 M parameters.

Keywords: real-time object detection; pomelo; transformers; hierarchical feature

Citation: Hou, B.; Li, G. HFD: Hierarchical Feature Detector for Stem End of Pomelo with Transformers. *Appl. Sci.* **2023**, *13*, 4976. https://doi.org/10.3390/app13084976

Academic Editors: Yujin Lim and Hideyuki Takahashi

Received: 2 March 2023
Revised: 8 April 2023
Accepted: 13 April 2023
Published: 15 April 2023

1. Introduction

Belonging to the genus Citrus of the family Rutaceae, pomelo (*Citrus grandis* L. Osbeck) is one of the three basic species of citrus cultivars, which account for approximately 25% of the output of Citrus fruit in China [1]. Pomelo is fragrant, sweet and sour, cool and moist, rich in nutrition, and high in medicinal value. It is not only a fruit that people like to eat, but also one with therapeutic effects [2].

Nowadays, most of the fruit detection methods consist of traditional image processing methods, which require hand-crafted features for various situations. It takes much effort and time to design those features [3]. In traditional image processing, the surface flaw of pomelo can be easily detected, but the stem end of pomelo is also drastically mistaken as a flaw. In recent years, deep learning has become more and more influential in the field of computer vision. With the progress of deep learning technology, image detection improves significantly.

Researchers optimize algorithms to accomplish vision-based tasks with high accuracy and reliability [4]. Deep learning approaches, especially vision transformer, can better perform computer-vision-related tasks [5]. Deep learning algorithms are stronger than traditional image methods for fruit detection [6]. They excel in feature representation and extraction, especially in automatically obtaining features from images [7]. Thanks to their powerful capabilities and easy assembly, they can solve complex and large problems more efficiently [8].

For the detection of the stem end of pomelo, there are no standard or even clear detection and grading guidelines. Researchers usually determine the detectors by experience. The deep learning method is good at extracting the hidden information from labeled image datasets [9].

Thus, this paper takes the detection of the stem end of pomelo as the research background and uses the deep learning method to build a detection transformer network that

meets the real-time requirements in the pomelo sorting system and improves the accuracy of pomelo detection.

To construct better deep learning architectures, we propose a hierarchical feature detector with transformers. The proposed model comprises the combination attention module (CAM) and the hierarchical feature fusion module (HFFM). For object detection, CAM can let any Vision Transformer (ViT) [10] variant append the patch tokens [PatTok] [11]. Therefore, this paper can integrate the Swin Transformer [12] backbone with CAM to be an object detector. The hierarchical feature detector (HFD) can obtain high scalability with the local attention of the Swin Transformer only using linear complexity.

We evaluated the effectiveness of the combination attention module and the hierarchical feature fusion module. We also compared the performance of HFD with other models, such as DETR [13], Deformable DETR [14], LSTM-SSD [15], SAAN-GRU [16], YOLO v4 [17], YOLO v5 [18], and YOLOS [11], as shown in Figure 1.

Figure 1. Capabilities of recent object detectors in terms of mean average precision (mAP) and frames per second (FPS).

The main highlights of this work include the following:

1. A transformer-based network model for instantaneous detection of pomelo was built, which has high precision and meets real-time requirements. Compared to some of the state-of-the-art models, our model shows the best performance on the pomelo dataset;
2. We designed the combination attention module, which is better for feature extraction in our dataset;
3. We designed the hierarchical feature fusion module, which can help detector obtain more accurate results;
4. A pomelo dataset was constructed to detect the stem end from the black spot.

The rest article of the article is organized as follows. Section 2 introduces some work related to detection methods, vision transformers, and detection transformers. Section 3 details the pomelo dataset. In Section 4, we specifically describe the HFD structure, including the CAM and HFFM. Subsequently, we designed a series of experiments to verify the effectiveness of our method in Section 5. Finally, Sections 6 and 7 provide a comprehensive review of our work and contributions.

2. Related Works

Before the advent of deep learning, the pomelo peel flaw detection task was usually carried out using machine learning. With the widespread use of deep learning, many fruit and vegetable detection algorithms have adopted a conjunction of traditional image algorithms and deep learning methods. Xiao et al. [19] used an improved feature fusion single multi-box detector for extracting RGB features for the detection of pomelo. The experimental results are good. However, their datasets are too small. This artificial neural network is only a detection function, and the generalization performance of the proposed model is not good. Huang [20] used a back-propagation neural network (BPNN) model to select the pomelo surface defects, pomelo shape, pomelo size, and other indicators. They built their own larger fruit dataset, and their data were mainly from daily shooting and the web. Li et al. [21] proposed using least-square support vector machine (LS-SVM) to identify pomelo on a 240-image dataset. They achieved good results with this small dataset. This machine learning method is applicable to the sorting of pomelo.

Moreover, for pomelo, some researchers even use infrared spectroscopy information [21,22]. Many traditional image algorithms are used to construct a system for pomelo maturity measurement and detection [22]. Such works are comprehensive. To determine categories, researchers count the pomelo color histograms and use thermal cameras to detect defects. Undoubtedly, these methods increase the hardware cost of a model that uses only cameras. The study by Jie et al. [23] shows that the conventional convolution neural network (CNN) achieved the best accuracy compared with the LS-SVM and BPNN for citrus grandis granulation determination. The quality of the detection model depends on the feature extraction. To improve the performance of CNN, they added the batch normalization layer. The detection model achieved 97.9% accuracy on the validation set. They point out that bands of 807–847 nm, 709–750 nm, and 660–721 nm are the spectra greatly related to pomelo granulation through analyzing the well-trained model layer by layer. Combined with some studies on functional groups, it is possible to speculate the change in internal substances, which may provide some hints to develop granulation-detecting equipment for pomelo.

The limitations of the current state of the art that motivate the present study lie in the small size of the number of pomelo datasets and the far less targeted improvement of the deep learning models.

2.1. Detection Methods

There have mainly been two kinds of detectors since the advent of deep learning. They are the one-stage detection framework and the two-stage detection framework [24,25]. The two-stage detection framework, which is represented by RCNN [26] and Fast RCNN [27], generates a series of sparse candidate boxes through CNN, and then classifies and regresses these candidate boxes. It has a more complicated training process because of the multistage complex pipeline. In practical applications, the time of inference is very long [24]. Theoretically, it is difficult for us to optimize. RCNN [26] uses CNN networks to extract image features from empirically driven artificial feature paradigms histogram of oriented gradients and scale invariant feature transform to data-driven representation learning paradigms to improve feature-to-sample representation. Fast RCNN [27] only performs feature extraction for the whole image full region once, introduces suggestion frame information, and extracts the corresponding suggestion frame features.

By comparison, one-stage detection framework (Representative YOLO [28], SSD [29], etc.) can avoid the problems mentioned above. YOLO [28] uses the whole image as the input of the network and takes target detection as a regression problem to solve it. YOLO directly regresses the position and category of the preselection box on the output layer. SSD [29] extracts feature maps of different scales for detection. Large-scale feature maps (the feature map in the front) can be used to detect small objects, while small-scale feature maps (the feature map in the back) can be used to detect large objects. Moreover, SSD uses prior boxes (default boxes) with different scales and aspect ratios.

In summary, one-stage detection frameworks detect objects in a single pass through the network. Two-stage detection frameworks use a two-stage process to detect objects. In the first stage, the network proposes regions of interest (ROIs) where objects may be located. In the second stage, the network classifies the proposed ROIs and refines their bounding boxes. One-stage detectors are faster and easier to use, but they sacrifice accuracy. Two-stage detectors are more accurate but are slower and more complex. In practical applications, provided that the real-time requirements are satisfied (FPS > 50), both one-stage and two-stage detection frameworks are suitable for distinguishing the stem end of pomelo from its black spots with higher accuracy.

2.2. Vision Transformers

The original ViT [10] is a model for image classification that uses a transformer-like architecture on various parts of the image. An image is processed as a series of small patches by transformers, making it easy to consider the interaction between patches at all positions, such as global attention. ViT [10] contains three main components: patch embedding, feature extraction from stacked transformer encoders, and classification head. However, due to the high computational complexity (increasing in a quadratic way with the image size), the original ViT cannot be easily applied to a wide range of visual tasks. By introducing the concept of a shifted window that supports patch reduction and local attention operations, the Swin-Transformer [12] mitigates the complexity problem and improves the adaptability to intensive prediction tasks (such as object detection). Pooling-based vision transformer [30] is able to reduce the ViT structure size and improve the spatial interaction ratio of ViT by controlling the self-attentive layer. A few methods use vision transformers as detector backbones. However, they achieve limited success [11,12,30].

2.3. Detection Transformers

Combining the structures of convolutional neural network backbones and transformer encoder–decoder, detection transformers discard the precisely designed components, such as anchor generation and maximum suppression. The study by Song et al. [31] shows that detection transformers can be effective detectors by configuring the attention module and refining the decoder. Compared to previous detectors [26–30], the original DETR [13] achieves accurate detection results, but the convergence speed is slow. For example, the Faster R-CNN [28] requires only 50 epochs for training while DETR needs 150 epochs. In order to solve this problem, Zhu et al. [14] propose Deformable DETR, which contains deformable attention to accelerate the slow training speed of DETR and utilize multi-scale features in the image.

3. Background

3.1. Pomelo Sorting System

As shown in Figure 2, the pomelo images used in this work were collected from a micro-diameter high-performance pomelo sorting machine developed by the Institute of Microelectronics of Chinese Academy of Sciences and Jiangxi Reemoon Sorting Equipment Co., Ltd., (Jiangxi, China). The machine vision part consists chiefly of high-resolution industrial cameras, LED warm light sources for providing sufficient light to the camera, a photoelectric switch which is used to control image capture, and conveyor belts with rollers.

The pomelo triggers the photoelectric switch to capture images by the cameras which have 1280 × 1024 resolution and a rate of up to 60 frames per second. When the cameras capture images, the pomelo rotates with the roller to obtain the information of the whole surface of one pomelo. As shown in Figure 3, the pomelo region is extracted after applying preprocessing methods to every image.

Figure 2. The pomelo sorting machine. Rollers on the conveyor belt can rotate the pomelo.

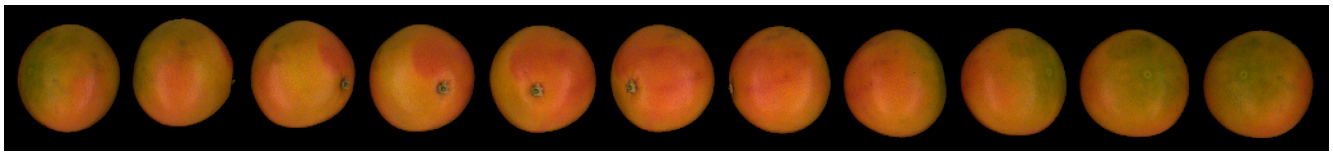
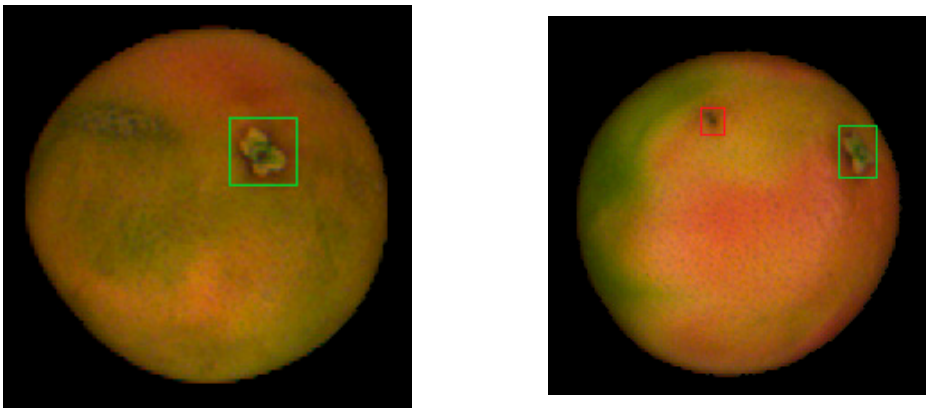


Figure 3. The eleven images captured while the pomelo rotating.

3.2. Dataset

From the machine vision part, we collected 11,253 pomelo images. Then, we marked each image with the stem end and the black spot in MS COCO (Microsoft Common Objects in Context) format. Based on the minimal tag of the category, we extracted the dataset to balance every object in the dataset.

There are 5173 images of pomelo labeled with 3561 stem ends and 3893 black spots as experimental data. As shown in Figure 4, we marked each detection object of pomelo with a bounding box. Nine-tenths of these images (4656) were randomly used as the training set, and the remaining images (517) were selected as the test set. The details about this dataset are given in Table 1.



(**a**) The green box annotates the stem end. (**b**) The red box annotates the black spot.

Figure 4. The annotated detection objects.

Table 1. Basic information on the pomelo dataset.

	Training Set	Test Set	Total
Stem end	3193	368	3561
Black spot	3481	412	3893
Total	6674	780	7454
Images	4656	517	5173

4. The Proposed Model

The hierarchical feature detector (HFD), illustrated in Figure 5, reconfigures the Swin Transformer's self-attention model. In this way, independent object detection can be supported, and parameters of Swin Transformer are fully reused. Through the detection heads (shown in the right part of Figure 5) for classification and box regression, the output embeddings of the decoder are used as final predictions.

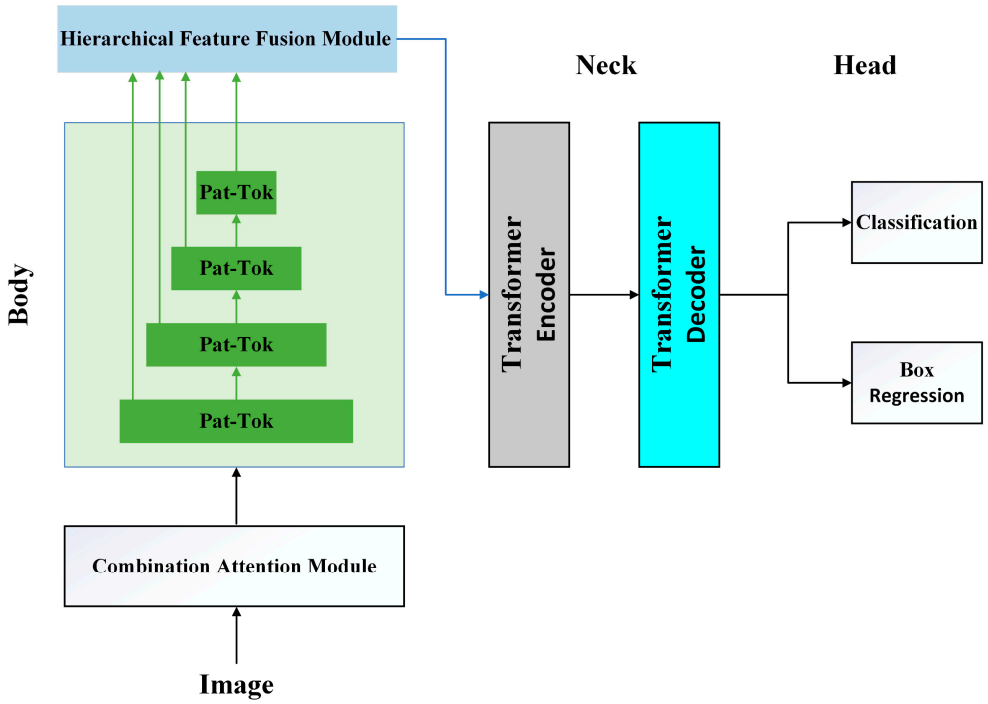


Figure 5. HFD architecture overview. Pat-Tok refers to the embedding of a flattened image patch.

4.1. Combination Attention Module (CAM)

Figure 6 represents the detailed structure of CAM, which has a novel attention module. We apply the efficient schemes in the Swin Transformer only to [Pat-Tok] × [Pat-Tok] attention based on the decomposition, which is the largest part of computational complexity. This adjustment totally reuses all the parameters of the Swin Transformer though the projection layers.

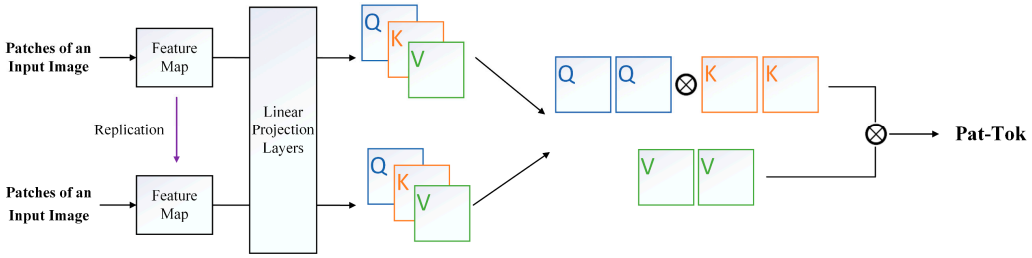


Figure 6. The structure of CAM. Q: query, K: key, V: value.

The fundamental [Pat-Tok]s are graded step by step across the attention layers, so that they can accumulate the core contents in the global feature map, for example, a spatial form of [Pat-Tok], according to the weights of attention operations, which are computed by queries and keys. As for [Pat-Tok] × [Pat-Tok] attention, the Swin Transformer performs local attention on each window barrier, but the shifted window barrier in consecutive blocks bridges the windows of the previous layers, giving connections among barriers to obtain global information. A similar method is used to generate hierarchical [Pat-Tok]. So, we reduce the number of [Pat-Tok]s by a factor of 4 at each stage. In this way, the resolution of feature maps changes from $H/4 \times W/4$ to $H/32 \times W/32$ over the four stages, where H and W denote the height and width of the input pomelo image.

4.2. Hierarchical Feature Fusion Module (HFFM)

Without any processing, all the multi-scale [Pat-Tok]s of CAM are fed into the neck component of HFD. The [Pat-Tok]s are encoded. Then, they are decoded into embeddings of objects. The multi-scale [Pat-Tok]s are fused linearly by the multi-scale deformable attention of the encoder via weighted aggregation. However, only a few [Pat-Tok]s are sampled and accounted for computational efficiency. So, we propose a simple but efficient hierarchical feature fusion module (HFFM), which fuses all tokens with multiple scales non-linearly, as shown in Figure 7, before putting them into the encoder. Compared with the simple linear gathering, the proposed HFFM extracts integral information from feature maps at different scales more accurately.

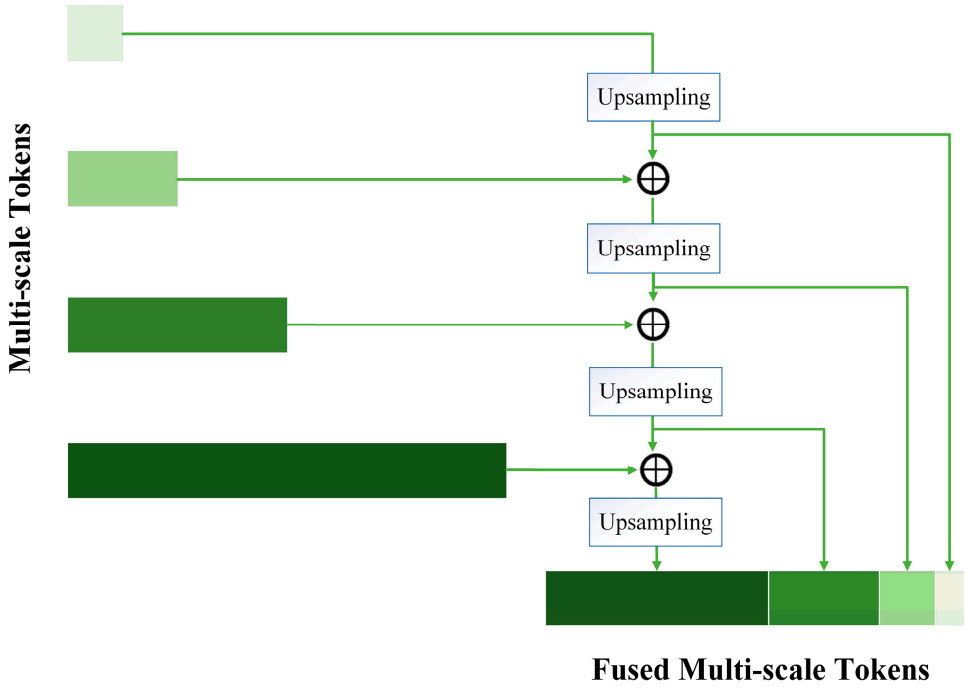


Figure 7. The structure of HFFM. HFFM mixes multi-scale [Pat-Tok]s into concatenated single output.

Specifically, all the multi-scale [Pat-Tok]s from the different stages are amassed to form feature maps at different scales with the same size (256) of channel dimension. They are mixed in a top-down manner. As illustrated in Figure 7, each operator receives two input feature maps for hierarchical feature fusion. The feature map in lower resolution is resized by the upsampling operator. Then, HFFM fuses the low-resolution feature map with the high-resolution one by bilinear interpolation. Hence, HFD can obtain fused multi-scale features, flattened with the spatial dimension, and chained for all scales as input to the encoder in the neck component of HFD.

4.3. Loss Function

Classification loss and box distance loss adopt the loss function of Deformable DETR. The Hungarian algorithm is used to find a bipartite matching between the ground truth box and the predicted box because the detection head of HFD returns a fixed-size set of bounding boxes, which is larger than the number of detection objects in a pomelo image. In general, there are four kinds of training loss functions: a classification loss l_{cla}, a box distance loss l_b, an IoU aware loss l_{IoU}, and a token labeling loss l_{token}.

$$l_{det} = \gamma_{cla} l_{cla} + \gamma_b l_b + \gamma_{IoU} l_{IoU} + \gamma_{token} l_{token} \tag{1}$$

For each training loss, the coefficient is set to be γ_{cla}= 1, γ_b= 5, γ_{IoU}= 2, and γ_{token}= 2.

$$l_{cla}(i) = -\log P_{c_i, a_i} \tag{2}$$

where c_i and a_i are the target class label and bipartite assignment of the i-th ground truth box.

$$l_b(i) = ||B_i, B_{a_i}||_1 \tag{3}$$

where B returns the largest box containing two given boxes.

Directly using the final [Pat-Tok] to predict the IoU score can increase detection confidence and mitigate the mismatch between ground truth and expected bounding boxes [32]. So, we predict the IoU score between the ground truth box $\hat{b}$ and predicted bounding one b by add a new FFN branch. The IoU aware loss is defined as

$$l_{IoU} = \frac{1}{B} \sum_{i=1}^{B} BCE\left([Pat - Tok]_i, IoU\left(\hat{b}, b\right)\right) \tag{4}$$

where BCE and B are binary cross-entropy loss function and the total number of objects in the input pomelo image.

Token labeling can solve multi-scale token recognition problems. HFD selects each [Pat-Tok] with an individual location-specific supervision, which is gathered by a machine analyst [33]. The token labeling loss is defined as

$$l_{token} = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{P^l} \sum_{i=1}^{P^l} F([Pat - Tok]_i, S_i) \tag{5}$$

where P^l and L are the numbers of tokens and scales in the feature map. F is the focal loss function [34]. $[Pat - Tok]_i$ returns the i-th [Pat-Tok] in the feature map from the body component of HFD. Corresponding to the i-th [Pat-Tok], S_i is the token-level label.

5. Experiment

On our pomelo dataset benchmark, we ran and evaluated all experiments. Then, we used the parameters mean average precision (mAP) and frames per second (FPS) to present model performance. The mAP is a performance metric commonly used in object detection tasks. It measures the average precision of a detector over multiple object classes. The mathematical expression for mAP is

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{6}$$

where N is the number of object classes and AP_i is the average precision for the i-th class. The average precision for each class is calculated as follows:

$$AP_i = \frac{1}{R_i} \sum_{j=1}^{R_i} \frac{P(j) * TP(j)}{P(j) * TP(j) + FP(j)} \tag{7}$$

where R_i is the total number of ground truth objects in class i, P(j) is the precision at the j-th retrieved object, TP(j) is the number of true positive detections at the j-th retrieved object, and FP(j) is the number of false positive detections at the j-th retrieved object.

Below, the implementation details of all experiments are introduced and we perform comparisons with state-of-the-art approaches.

5.1. Implementation Details

All models were trained using NVIDIA GeForce RTX 3090 cuDNN v8.0.05 with Intel(R) Xeon(R) Silver 4316 CPU @ 2.30 GHz. Applying mean values and standard deviations, we normalize the images into [0,1]. We also adopt some standard data augmentation methods, such as rotating and mirroring.

Using PyTorch, we train HFD with AdamW [35], where β_1 and β_2 are set as 0.9 and 0.99. HFD uses the initial learning rate of 10×-4 for its body, neck, and head. On the contrary, for the pre-trained body, neck, and head, Deformable DETR (Swin Transformer) and DETR are trained with the initial learning rate of 10×-5, which is the original setting of Deformable DETR. Following the YOLOS setting, YOLO v4, YOLO v5, and YOLOS are trained with the same initial learning rate of 10×-4.

Under our pomelo dataset, we uniformly set the number of training epochs to 150 and the batch size to 64, comparable with other state-of-the-art approaches. In addition to the HFD training from scratch, we use the best weights of pre-training to train other comparison models. More specifically, we set the anchor box size and scale by default. The detailed parameter settings and training code of the comparison model are open access [11,13–18]. During the initial epochs of HFD training, the validation accuracy and validation loss fluctuate rapidly as the model adjusts to the pomelo data. As the training progresses, the validation accuracy increases, while the validation loss decreases. This indicates that the model is improving its ability to accurately detect objects in pomelo images. During the final phase of training, validation accuracy tends to plateau and validation loss fluctuates in a very small range around 0.6.

5.2. Comparison with State of the Art

Obviously, as shown in Figure 1, the HFD achieves the highest mAP over other comparison models, and the inference speed meets the real-time requirements (FPS > 50). It is noteworthy that HFD exceeds YOLOv5-CSPDarknet53 by 1.59% with the highest mAP of 89.65%. The main reason is that the multi-head attention mechanism used in the transformers allows the model to focus on multiple regions of an image simultaneously. This makes the model more efficient at detecting objects across different scales and sizes and improves its ability to detect small objects. Although the accuracy of YOLOS is acceptable, the detection speed of YOLOS is slow because the computational complexity for attention in YOLOS is quadratic. With the same backbone, HFD presents better performance than DETR and Deformable DETR, both in terms of accuracy and speed. This is because CAM and HFFM enable the detector to perform the feature extraction ability of transformers more efficiently. Although the accuracy of HFD is higher than that of YOLOv5, the speed of HFD is slower than that of YOLOv5. The main reason is that YOLOv5 transforms the target detection problem into a single regression problem, which improves the detection speed. In practical applications, for the detection task of pomelo, accuracy is the primary goal of algorithm optimization when the real-time requirements are satisfied (FPS > 50). As shown in Table 2, the performance of HFD is optimal for both the total accuracy and the accuracy for pomelo stem end and black spot.

Table 2. Comparison of HFD with other network architecture.

Model	Backbone	Parameters (MB)	mAP	Speed (FPS)	Stem End	Black Spot
DETR [13]	Swin-tiny [12]	42.94	75.50	78.91	83.22	68.58
DETR [13]	Swin-small [12]	67.35	78.64	62.53	87.39	70.76
DETR [13]	Swin-base [12]	104.56	81.28	51.87	90.68	72.82
Deformable DETR [14]	Swin-tiny [12]	39.07	82.15	60.72	91.43	73.80
Deformable DETR [14]	Swin-small [12]	60.63	84.72	49.31	93.75	76.59
Deformable DETR [14]	Swin-base [12]	98.11	87.33	37.24	96.02	79.51
LSTM [15]	SSD [29]	**5.69**	72.54	31.10	80.71	65.29
SAAN-GRU [16]	ResNet-50 [36]	10.76	70.28	48.93	79.42	62.16
YOLO v4 [17]	CSPDarknet53 [37]	29.46	85.79	84.65	94.91	77.58
YOLO v5 [18]	CSPDarknet53 [37]	41.02	88.06	**103.58**	96.67	80.31
YOLOS [11]	DeiT-Ti [38]	7.13	76.24	45.19	81.84	71.20
YOLOS [11]	DeiT-S [38]	31.85	80.83	36.64	86.17	76.02
YOLOS [11]	DeiT-B [38]	102.79	85.47	29.42	93.56	78.19
HFD	Swin-tiny [12]	40.96	83.44	98.03	92.21	75.56
HFD	Swin-small [12]	61.17	86.29	82.86	95.48	78.04
HFD	Swin-base [12]	100.34	**89.65**	70.92	**98.23**	**81.93**

The best results in every category are marked in **bold**.

6. Discussion

We empirically demonstrate the combination attention module (CAM) and the hierarchical feature fusion module (HFFM) in a pomelo dataset for our model (HFD). Based on current experiments, the effects of different loss functions are also discussed.

6.1. Computational Complexity

For Swin Transformer [12], the computational complexity is $O\left(d^2S + dw^2C\right)$, where S denotes the number of tokens for self-attention, C denotes the number of tokens for cross-attention, d is the dimension of embedding, and w denotes the height and width of the window. As described in Section 4, CAM extends the Swin Transformer to be an object detector. The computational complexity of HFD is $O\left(d^2P + dw^2P\right)$, where P denotes the number of [Pat-Tok]s.

6.2. Model Architecture

To verify the effectiveness of the CAM, we refer to HFD-Swin-base. The [Pat-Tok] $\times$ [Pat-Tok] attention operation is removed. As shown in Table 3, the mAP of HFD-Swin-base without CAM is lower than that with CAM by 2.24%. However, HFD with CAM takes a little more time. Moreover, Table 3 shows that HFD has higher parameters than that without CAM.

Table 3. The effectiveness of the combination attention module for HFD with Swin-base.

Model	CAM	Parameters (MB)	mAP	Speed (FPS)
HFD-Swin-base	$\times$	93.06	87.41	73.68
HFD-Swin-base	$\surd$	100.34	89.65	70.92

For HFD, HFFM is added for better optimization. HFFM makes the detector better at extracting multi-scale features and this non-linear fusion is very simple and efficient.

With the Swin-base backbone, as shown in Table 4, the extension to HFD only adds 2.49 M parameters but mAP improves from 85.37 to 89.65. This is a significant performance gain for the better trade-off between accuracy and speed, while the runtime performance dropped by 5.28 FPS.

Table 4. The effectiveness of the hierarchical feature fusion module for HFD with Swin-base.

Model	HFFM	Parameters (MB)	mAP	Speed (FPS)
HFD-Swin-base	×	97.85	85.37	76.20
HFD-Swin-base	√	100.34	89.65	70.92

6.3. Loss Function

As illustrated before, HFD has four types of training loss: classification loss l_{cla}, box distance loss l_b, IoU aware loss l_{IoU}, and token labeling loss l_{token}. To further understand the roles IoU aware loss and token labeling loss play, we studied the impacts caused by different loss functions, which are shown in Table 5.

Table 5. Performance change with different loss function.

Model	I_{IoU}	l_{token}	Parameters (MB)	mAP
HFD-Swin-base	×	×	99.51	87.42
HFD-Swin-base	√	×	100.26	88.91
HFD-Swin-base	√	√	100.34	89.65

IoU aware loss and token labeling loss both contribute to the performance improvement. Although they make a decrease in the inference speed of the detector, this decrease is completely acceptable. Specifically, IoU aware loss helps the performance improvement of the detector more because for the pomelo detection task, there is a high probability of mismatch between ground truth and expected bounding boxes.

7. Conclusions

In this paper, to accurately and efficiently distinguish the stem end of pomelo from its black spot, we propose a hierarchical feature detector (HFD) model with the combination attention module (CAM) and the hierarchical feature fusion module (HFFM). Figure 8 shows the inference results of HFD.

On our pomelo dataset, HFD achieved mAP of 89.65% at 70.92 FPS with 100.34 M parameters; the mAP is 8.37% greater than DETR, which also has the backbone of Swin Transformer (base). It is competitive with the state-of-the-art transformer-based detectors such as DETR, Deformable DETR, and YOLOS and one-stage detectors such as YOLO v4 and YOLO v5 with high detection accuracy and real-time performance. In particular, the mAP of the stem end of the pomelo reaches 98.23% in HFD-Swin-base, and the mAP of the black spot reaches 81.93%.

The limitations of this paper are mainly that it reports some improvements of the object detection algorithm in the case of a specific fruit (pomelo). When it comes to possible future enhancements, the proposed HFD should be applied to a wider variety of fruits for object detection and have a wider practical application impact. The detector can continue to be optimized to achieve better accuracy and faster speed.

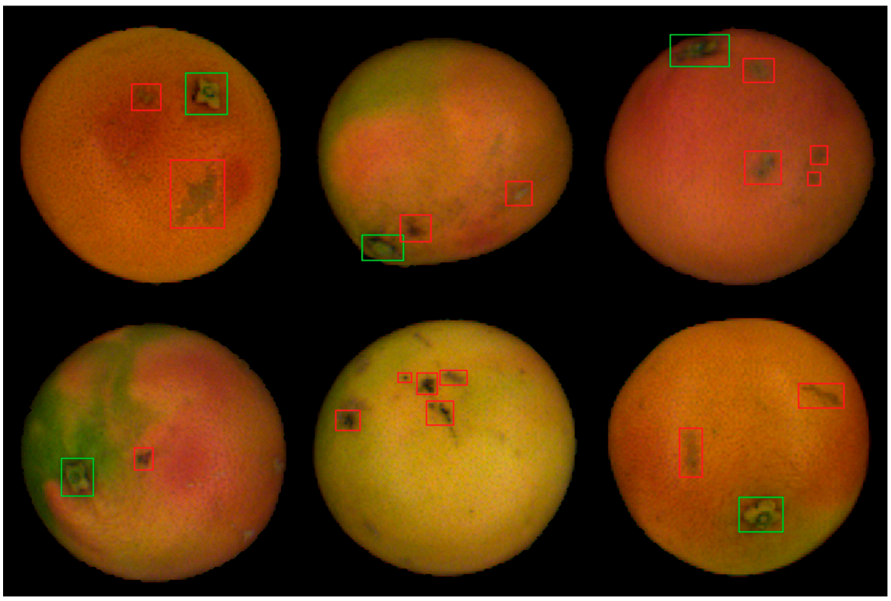


Figure 8. The detection results of HFD on pomelo. The stem end can be detected from black spots. Green and red boxes indicate the stem end and black spots respectively.

Author Contributions: Conceptualization, B.H.; methodology, B.H.; software, B.H.; validation, B.H.; formal analysis, B.H.; investigation, B.H.; resources, G.L.; data curation, B.H.; writing—original draft preparation, B.H.; writing—review and editing, B.H.; visualization, B.H.; supervision, G.L.; project administration, G.L.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the International Partnership Program of Chinese Academy of Sciences (no. 184131KYSB20200033), the National Key R&D Program of China (no. 2018YFD0700300), and the Chinese Academy of Sciences Engineering Laboratory for Intelligent Logistics Equipment System (no. KFJ-PTXM-025).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest that could be perceived as prejudicing the impartiality of the research reported.

References

1. Xie, R.; Li, G.; Fu, X.; Wang, Y.; Wang, X. The distribution of main internal quality in pummelo (*Citrus grandis*) fruit. *AIP Conf. Proc.* **2019**, *2079*, 1026–1034.
2. Li, X.; Xu, S.; Pan, D.; Zhang, Z. Analysis of Fruit Quality and Fuzzy Comprehensive Evaluation of Seven Cultivars of Pomelos. *J. Anhui Agric. Sci.* **2016**, *44*, 78–80.
3. Kamilaris, A.; Prenafeta-Boldu, F. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
4. Balakrishnan, A.; Ramana, K.; Ashok, G.; Viriyasitavat, W.; Ahmad, S.; Gadekallu, T. Sonar glass—Artificial vision: Comprehensive design aspects of a synchronization protocol for vision based sensors. *Measurement* **2023**, *211*, 112636. [CrossRef]
5. Ramana, K.; Srivastava, G.; Kumar, M.; Gadekallu, T.; Lin, J.; Alazab, M.; Iwendi, C. A Vision Transformer Approach for Traffic Congestion Prediction in Urban Areas. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3922–3934. [CrossRef]
6. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. Acm.* **2017**, *60*, 84–90. [CrossRef]
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
9. Sun, X.; Ma, L.; Li, G. Multi-vision Attention Networks for on-Line Red Jujube Grading. *Chin. J. Electron.* **2019**, *28*, 1108–1117. [CrossRef]

10. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
11. Fang, Y.; Liao, B.; Wang, X.; Fang, J.; Qi, J.; Wu, R.; Niu, J.; Liu, W. You Only Look at One Sequence: Rethinking Transformer in Vision through Object Detection. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 26183–26197.
12. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
13. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 213–229.
14. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv* **2021**, arXiv:2010.04159.
15. Zhu, M.; Liu, M. Mobile Video Object Detection with Temporally-Aware Feature Maps. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 5686–5695.
16. Xiao, Z.; Qi, J.; Xue, W.; Zhong, P. Few-Shot Object Detection with Self-Adaptive Attention Network for Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4854–4865. [CrossRef]
17. Bochkovskiy, A.; Wang, C.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Seattle, WA, USA, 13–19 June 2020.
18. Jocher, G. yolov5. 2021. Available online: https://github.com/ultralytics/yolov5 (accessed on 16 February 2023).
19. Xiao, D.; Cai, J.; Lin, S.; Yang, Q.; Xie, X.; Guo, W. Grapefruit Detection Model Based on IFSSD Convolution Network. *Trans. Chin. Soc. Agric. Mach.* **2020**, *51*, 28–35.
20. Huang, J.; Liu, Y.; Yang, D. The Classification of Grapefruit Based on BP Neural Network. *Hubei Agric. Sci.* **2018**, *57*, 112–115.
21. Li, X.; Yi, S.; He, S.; Lv, Q.; Xie, R.; Zheng, Y.; Deng, L. Identification of pummelo cultivars by using Vis/NIR spectra and pattern recognition methods. *Precis. Agric.* **2016**, *17*, 365–374. [CrossRef]
22. Shang, J. Progress of Nondestructive Determination Technologies Used in Grapefruit Classification. *Mod. Food* **2018**, *3*, 60–62.
23. Jie, D.; Wu, S.; Wang, P.; Li, Y.; Ye, D.; Wei, X. Research on Citrus grandis Granulation Determination Based on Hyperspectral Imaging through Deep Learning. *Food Anal. Methods* **2021**, *14*, 280–289. [CrossRef]
24. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
25. Agarwal, S.; Terrail, J.; Jurie, F. Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks. *arXiv* **2018**, arXiv:1809.03193.
26. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
27. Girshick, R. Fast R-CNN. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal Canada, 7–12 December 2015; pp. 91–99.
29. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
30. Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; Oh, S. Rethinking spatial dimensions of vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2021; pp. 11936–11945.
31. Song, H.; Sun, D.; Chun, S.; Jampani, V.; Han, D.; Heo, B.; Yang, M. An Extendable, Efficient and Effective Transformer-based Object Detector. *arXiv* **2022**, arXiv:2204.07962.
32. Wu, S.; Li, X.; Wang, X. Iou-aware single-stage object detector for accurate localization. *Image Vis. Comput.* **2020**, *97*, 103911. [CrossRef]
33. Jiang, Z.; Hou, Q.; Yuan, L.; Zhou, D.; Shi, Y.; Jin, X.; Feng, J. All tokens matter: Token labeling for training better vision transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 18590–18602.
34. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
35. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
37. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
38. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 10347–10357.