Multimedia Tools and Applications (2024) 83:7811–7825
https://doi.org/10.1007/s11042-023-15938-1

Apple ripeness identification from digital images using transformers

DONE

ComfyUI version?

Bingjie Xiao[1] · Minh Nguyen[1] · Wei Qi Yan[1]

Received: 2 February 2022 / Revised: 29 March 2023 / Accepted: 29 May 2023 /
Published online: 10 June 2023

Abstract
We describe a non-destructive test of apple ripeness using digital images of multiple types of apples. In this paper, fruit images are treated as data samples, artificial intelligence models are employed to implement the classification of fruits and the identification of maturity levels. In order to obtain the ripeness classifications of fruits, we make use of deep learning models to conduct our experiments; we evaluate the test results of our proposed models. In order to ensure the accuracy of our experimental results, we created our own dataset, and obtained the best accuracy of fruit classification by comparing Transformer model and YOLO model in deep learning, thereby attaining the best accuracy of fruit maturity recognition. At the same time, we also combined YOLO model with attention module and gave the fast object detection by using the improved YOLO model.

Keywords YOLO · Transformer · Object detection

1 Background

With the deepening of deep learning research, a spate of deep learning models appear in the field of natural language processing (NLP) as one of the essential directions of artificial intelligence. Compared with conventional machine learning methods, the advantages of deep learning are mainly in that the complex feature extraction process is not required. Therefore, crucial deep learning structures such as convolution neural networks (CNN or ConvNet) and recurrent neural networks (RNN) have been boardly employed in natural language processing (NLP) in recent years and have achieved significant progress [17, 18]. Visual object classification is able to reduce the consumption of human labor through computer vision which is an essential work of digital image and video processing as well as computer vision. Because of the extensive use of deep learning, visual object classification algorithms have been updated and prompted quickly [8, 12].

!!

Visual object detection is harnessed in face detection, vehicle detection, self-driving cars, pedestrian counting, security systems, etc. [14, 15, 22]. This paper is to utilize the characteristics of visual object detection to realize object detection of fruit maturity [35, 36].

✉ Wei Qi Yan
dcsyanwq@gmail.com

1 Auckland University of Technology, Auckland, New Zealand

The labor shortage in New Zealand when the fruit is ripe has inspired our deep learning applications. We need to construct a model so that the model can automatically classify the shape, size, and maturity of fruit, so as to help agriculture implement the automatic picking of robotic arms. Our project aims to make use of the Transformer model from natural language processing combined with deep learning models to achieve the classification of fruits with different classes of fruit ripeness. In this paper, we extensively investigate fruit classification methods in indoor and outdoor environments, choose apples and pears as the experimental targets. We are use of mobile phones to take pictures of fruits and generate the images for our dataset. We classify the fruits in the supermarkets, and distinguish the ripeness of various fruits, such as ripe apple, overripe apple, ripe pear, overripe pear. In our experiments, we mainly introduce two types of object detection methods, i.e., YOLO and Transformer [2, 3, 5].

As shown in Fig. 1, we take use of rectangular boxes to locate fruits in the image and label the classes and maturities (i.e., ripe apple, overripe apple, ripe pear, overripe pear). Then, we train the model by using Transformer encoder and decoder, get the prediction results of fruit category and maturity. The dotted lines in Fig. 1 show the block of the Transformer, which is actually a process of training the detector. In the YOLO model, we make use of the blocks of YOLO to train the detector. Figure 1 shows a complete experimental procedure. We are use of PyTroch platform for experiments. After the data is marked, it is sent to the model for our experimentation.

We will introduce our Transformer model and its implementation for our experiments in the second part of this article. The third part of the paper will introduce the specific process of the experiment. We show the experimental results in the fourth part. Finally, we conclude and envision future work at the end of the paper.

2 Related work

Natural Language Processing (NLP) is one hotspot of artificial intelligence research. NLP performs tasks such as translation, semantic analysis, chatbot development, text mimicry, and text-to-speech [28]. Deep learning provides multilayer perceptron models to train deep

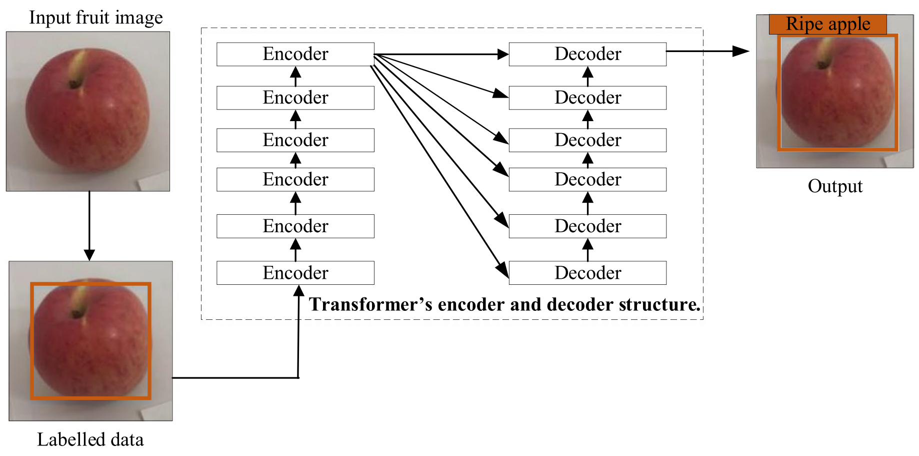

Fig. 1 The workflow of fruit ripeness recognition using Transformer model

nets with abstract data and to discover complicated structures in big data. Therefore, nowadays, the primary task of visual object detection is still a very challenging job, with a great potential and a large room for betterment [28, 31].

In the practical application of digital images, object detection is a complex image retrieval guided by visual saliency. In real life, our human eyes can detect the saliency of complex scenes by using the attention selection mechanism [34, 38], and accurately find the target of interest through semantic analysis [18, 25]. The object detection is also based on the retrieval performance problems existing in the existing image retrieval system, through drawing on the attention selection mechanism of the human visual system to establish a visual saliency calculation model. At present, the low-level visual features such as color and texture extracted by the most of image content retrieval systems are quite different from the semantic features by humans in image understanding, resulting in the problem of semantic gap in retrieval. However, Transformer object detection establishes a computational model of attention selection mechanism from the bottom up, making the model consistent with the saliency model of the human visual perception system. Visual object detection is based on the saliency image feature extraction and matching strategy, the combination of local features and global features for feature description is probed, and finally an image retrieval method combining top-down attention selection and bottom-up attention selection is realized.

Fruit detection is based on the inspiration of New Zealand's demand for smart agriculture. With the rapid development of visual object detection, fruit detection methods can now achieve high efficiency and high precision.

The "location + classification" issue is related to pattern classification for visual object detection, which simply goes from visual object classification to the position of visual object, and further to the class label and position of multiple objects [11, 26, 29]. Fruit harvesting robots [20, 30] have been developed by using YOLOv3 and YOLOv5. The undetected apples of the YOLOv3 model only have 9.2% of the entire dataset, while YOLOv5 can accurately detect fruits and only 2.8% of undetected apples [19, 23, 30].

Tang [29] et al. proposed a coordinate-based anchor-free (CBAF) [24] module for visual object detection. The output coordinates are directly predicted by using SSD or a branch of RetinaNet [7]. The focus is on visual object detection of small target objects. For our experiments, it is impossible to have only a single apple or pear in the images while collecting data. Usually, in our dataset, apples may be on trees or pears in fruit bowl, how to accurately extract small objects in the image is a problem [16]. An algorithm based on an improved YOLOv5 [10] is offered to solve the problem of loss rate of small visual objects and weak classifiers. The characteristics of the small target datasets are applied to optimize the deep learning network, increase the residual network, converge the algorithm speed to prevent the gradient from small or vanishing, and improve the detection outcome of the model to prevent overfitting [26].

The difficulty of NLP is that most NLP algorithms are based on complex deep neural networks, such as RNNs, LSTMs, and GRU models. Alkalouti and Masre [4] combined YOLO model and LSTM for visual object detection. For our experiments, how to adjust the YOLO model so that the model adapts to our experiments, thereby, how to improve the detection speed and accuracy is one of the current experimental goals.

Dai [9] et al. made improvements to the Detection Transformer (DETR) model for large objects rather than small ones. The backbone of R-CNN can be frozen, a

DETR is referenced that can find object localization and preferences in the task, which extends it to object query shuffle and attention mask so as to carry out multi-query localization [37]. The relevant experiments [21] proved that DETR has limitations in visual object detection. In contrast, YOLOv5 can detect visual objects faster and accurater.

Arkin [6] et al. analyzed the pros and cons of CNN models and transformer models. CNN extracts visual feature information from the local to the global through convolutional layer, while the transformer model directly grasps global information and then makes use of each patch to achieve self-attention. Transformer does not have bionic features, Transformer needs data samples to achieve better results under the same parameters. However, the transmission result of the Transformer is better than that of R-CNN [33]. How to adjust the parameters to combine the advantages of the two models is also a problem to be considered in our experiments.

Abozeid [1] et al. improved Swin Transformer model, comprising of an encoder, a decoder, and skip connections. For outdoor weather changes due to ambient noises, we experiment with object stacking, computation of objects, and upsampling the feature size with a dedicated patch [32].

Hendria [13] integrates Swin Transformer and CNN models, by using Non-Maximum Suppression (NMS), soft-NMS, Non-Maximum Weighting (NMW), and weighted box fusion (WBF) through combining predictions from multiobject detection models [27].

We selected YOLOv5 model for deep learning DETR and Swin Transformer model and train our model by comparing the advantages and disadvantages between different models. YOLO model has the characteristics of fast detection speed and high accuracy. The Transformer makes use of the advantages of self-attention to avoid the limitations of conventional CNN-based or RNN-based models, which is easy to combine multimodal data and provide greater flexibility for the model.

3 Methodology

3.1 YOLO

The series of YOLO algorithms segment the input image into grids, each grid is only responsible for visual objects whose center points are fallen into the grid and marked with the height and width of the bounding boxes. If the point of an object is fallen into a grid, then this grid corresponds to the detecting object.

Regarding the output of deep neural networks, three bounding boxes are generated for each grid cell. The output of each bounding box has three parameters: The box parameter of the object, which has four values in total, are represented by (x, y, h, w). Confidence is a probability within an interval [0,1.0]. The confidence of the bounding box equals to the probability of the object multiplied by the IOU of the bounding box and the actual bounding box of the visual object as shown in Eq. (1).

$$C_i^j = P_r(Object) \times IOU_{pred}^{truth} \tag{1}$$

where conditional class probabilities are all values in the interval [0,1.00]. In YOLOv5, C_i^j represents the area estimation of the detection target. $P_r(Object)$ is the probability that has an object in the bounding box. If there is an object, $P_r(Object)$ equals to 1.0, otherwise $P_r(Object)$ equals to 0. IOU_{pred}^{truth} is the ground truth and predicted box cross-combination ratio

Apart from the basic composition of the YOLO series, the YOLO network has also undergone various changes from YOLO basis model to YOLOv5.

YOLOv5 initializes with 9 anchors, which are employed to the three feature maps, respectively. Each grid cell of each feature map has three anchors for prediction. In the YOLOv5 model, the feature map with a larger scale is closer to the front, and the downsampling rate of the original image is smaller, and the receptive field is smaller. Therefore, it is relatively possible to predict visual objects with a smaller scale (small targets), and the assigned anchors are smaller. The smaller the scale of the feature map, the higher the downsampling rate related to the original image, and the larger the receptive field is, larger scale objects (large objects) can be predicted. The detection rates of YOLOv5 vary with the size of the feature map.

YOLO takes use of the mean squared error as the loss function to optimize the model parameters; namely, the mean squared error of the $S \times S \times (B \times 5 + C)$ dimensional vector output by using the vector of the corresponding real image as shown in Eq. (2)

$$loss = coordError + iouError + classError \tag{2}$$

where coordError represents the coordinate error between the predicted and real data, iouError shows the IoU error, and classError displays the classification error. The regression of YOLOv5 roughly has a region range, and then adjusts the bounding box of the candidate regions to be closer to the bounding box. The deviation of adjusting the bounding box of the candidate region to be closer to the real bounding box is the translation from the offset object position to the anchor point position.

INTERSECTION OVER UNION

The grid-based offset indicates that the anchor location is fixed, the offset is equal to the translation between the object position and the anchor position, where p_w and p_h are denoted as the width and height which are applied to predict the offsets directly t_w and t_h. The offset is shown as Eqs. (3), 4, 5 and (6):

$$t_x = log\left(\frac{bbox_x - c_x}{1 - (bbox_x - c_x)}\right) \tag{3}$$

$$t_y = log\left(\frac{bbox_y - c_y}{1 - (bbox_y - c_y)}\right) \tag{4}$$

$$\mathrm{t_w} = \log\left(\frac{\mathrm{gt_w}}{\mathrm{p_w}}\right) \tag{5}$$

$$\mathrm{t_h} = \log\left(\frac{\mathrm{gt_h}}{\mathrm{p_h}}\right) \tag{6}$$

where c_x and c_y stand for rectangular coordinates, t_x, t_y, t_w, t_h indicate the offsets to be predicted, b_x, b_y, b_w, b_h represent the final results obtained by using the proposed model, respecitvely. The pseudocode of our algorithm is listed as follows.

```
Input: An apple image
Initialization;
for $x$ in interval (0, width)
for $y$ in interval (0, height)
For $w_c$ in interval (1, 2000)
For $h_c$ in interval (1, 2000)
        For a feature map $img_c$
        $img_c$ = $img$[$x$-$w_c$/2:$x$+$w_c$/2, $y$-$w_c$/2:$y$+$w_c$/2];
        $confidence$ = $classifier$ ($img_c$)
        If $class$ == "ripe apple" $detect_{result}$ ([$confidence$])
        Endfor
Endfor
Endfor
Endfor
Endfor
Output: Apple detection result.
```

A loop is applied to show the iteration process of the entire model. In this algorithm, we input the entire image, then extract the feature map, execute the regression algorithm, and obtain the classification output.

3.2 Transformer

The attention mechanism is an essential module in the transformer model. The attention mechanism generates a mask through the operations. The mask is scored to evaluate the fruit that needs to be recognized. As shown in Fig. 2, first of all, the model needs to perform an embedding operation on the input data. After the embedding is completed, it will be the input to the encoder layer. After the self-attention layer processes the data, the data is sent to the feedforward neural network. The calculation of the feeforward neural network can be parallelized to obtain the output. The output will be input to the next encoder.

The essence of natural language processing is vectors, namely, converting images into vectors. The self-attention input is the entire sequence, and the input is also a sequence, which is the same as the input sequence. Each output vector must take into account the information of the entire input sequence, so self-attention introduces three matrices query, key, and value. We convert the dataset images to vectors according to Eqs. (7), (8) and (9).

Query
$$\mathbf{Q} = \mathbf{X} \times \mathbf{W_q} \tag{7}$$
Self Attention

key, relevance
$$\mathbf{K} = \mathbf{X} \times \mathbf{W_k} \tag{8}$$

Value, information being queried
$$\mathbf{V} = \mathbf{X} \times \mathbf{W_v} \tag{9}$$

Weights adjust To fresh

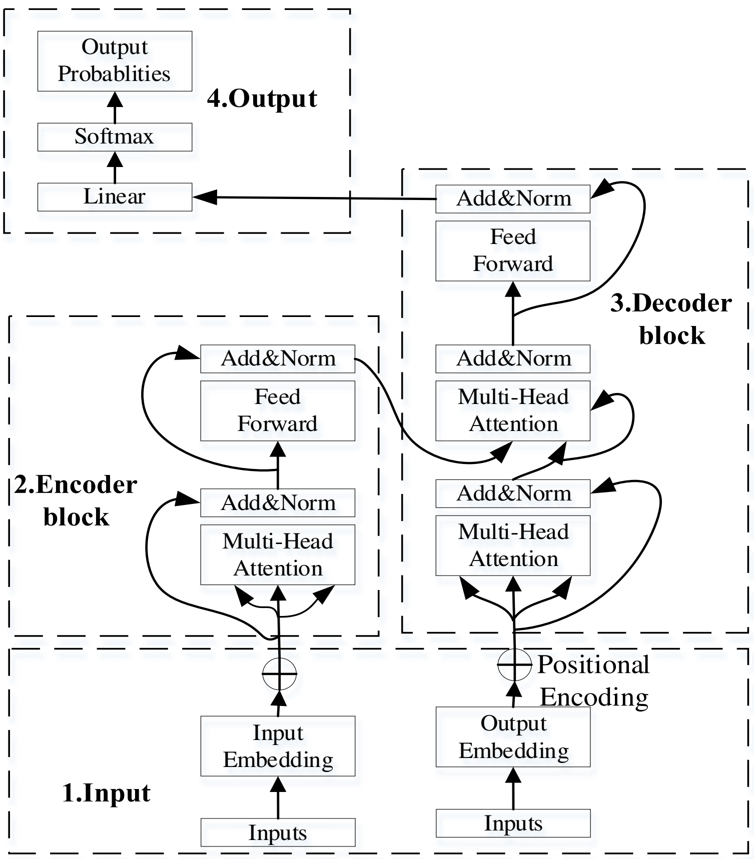

Fig. 2 Transformer structure

where **Q** represents the query vector, **K** (key vector) means a vector representing the relevance of the queried information and other information, **V** (value vector) shows the vector of the information being queried. The attention weight of X corresponding information **V** is proportional to **Q** multiply **K**. It is equivalent to saying: Attention weight is determined by **X** itself, so it is called self-attention. $\mathbf{W_q}$, $\mathbf{W_k}$, and $\mathbf{W_v}$ will be updated and changed according to the task goal, ensuring the effectiveness of the self-attention mechanism. **Q, K**, and **V** form the inner product of vectors, which means that the angle between two vectors is represented, and the projection of one vector on another vector. A large projection value means that the correlation between the two vectors is high. The meaning of softmax is normalization. The row vector $\mathbf{K}^T$ in Eq. (10) indicates the inner product with itself and other two row vectors respectively, and shows the result of the inner product operation between each vector and itself as well as other vectors. As Transformer limits the computation of attention to each window, thereby we reduce the amount of computation as shown in Eq. (10):

Transpose of relevance

$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = Softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{B}\right)\mathbf{V} \tag{10}$$

where **B** represents bounding box position, *T* means matrix transpose, the relative position coding is added to **Q** and **K** in the Swin Tansformer model.

3.3 Detection transformer,swin transformer and CornerNet

Detection Transformer (DETR) was employed for visual object detection or panoramic segmentation. This is the first object detection framework that successfully integrates Transformer as the central building block of the detection model. Compared with previous detection methods, DETR effectively eliminates the need for hand-designed components, such as non-maximum suppression (NMS) program, anchor point generation, and so on. DETR is a straightforward end-to-end framework.

In Fig. 3, the input of the network is a 3-channel RGB image wherer the backbone is CNN, which extracts the features, and then combines the position information and input it into the encoder and decoder of the transformer model to obtain the detection results of the model. Each output is a bounding box, where each box represents a tuple, including the category of the object and the position of the detection box.

The image input by using Swin Transformer is subjected to a convolutional layer for patch mapping, and each small block is mapped into a pixel to expand the channel. The feature map is initially input to a stage, which comprises two layers of transformers. There is a pooling operation between stages to reduce the size of the data to be processed, from the initial local information search to the extraction of global information.

In Fig. 4, black box shows patch, red box represents a window perform self-attention. Window attention is applied to divide the image into different windows according to a given size. Each time, the attention of the Transformer is only calculated inside the window. At this point, we see that the Swin Transformer and ResNet are designed with a network with noticeable hierarchical results. The structure at the bottom tackles more and more local data, and the network at the top processes fewer data but more semantic information. The difference is that Swin Transformers mainly takes advantage of Transformers to extract information, while ResNet takes use of a convolution kernel.

Like YOLOv5 and Transformer, CornerNet also use of a backbone network. CornerNet is an anchor-free network, which simply expresses the anchor box as a key point symmetrical between the upper left corner and the lower right corner, then adopts a new pooling layer corner pooling to find diagonal points better.

The CornerNet network is use of the Hourglass network as the backbone network, followed by two prediction models to predict the upper-left corner and lower-right corner of the box respectively. Each channel of CornerNet's heatmap is a binary mask, which

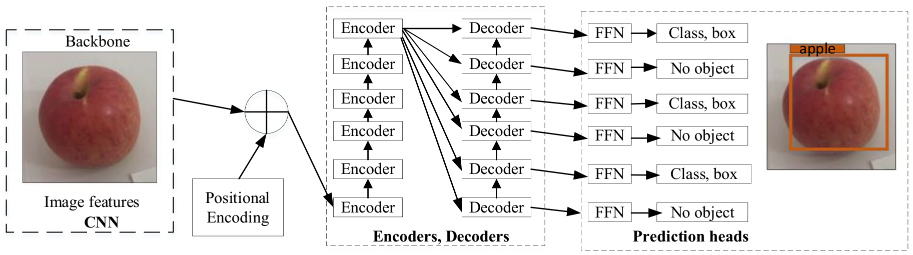

Fig. 3 DETR structure

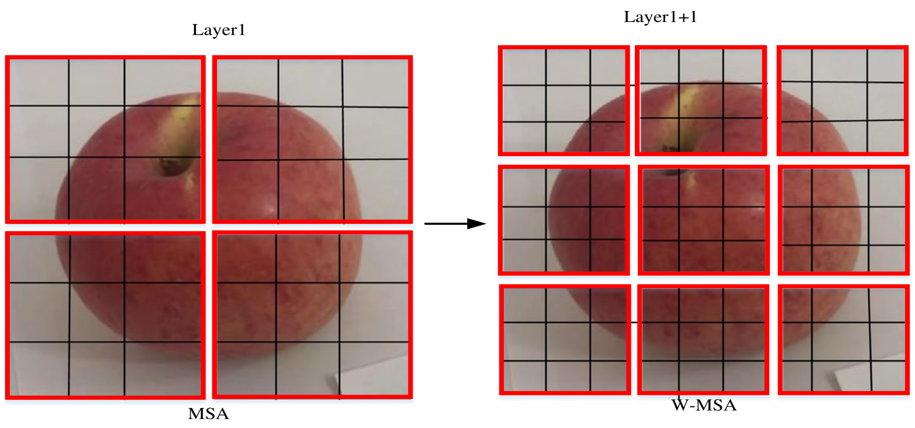

Fig. 4 A sample of window attention and shift window attention

predicts the position of each object key point pair, and each corner point makes a distance prediction with other corner points as an embeddings vector, and then similar embeddings or belong to the same distance between them, if a box is small, the anchor box will be generated and the predicted result will be obtained.

3.4 Experiment setting and evaluation method

We have utilized a mobile camera to take 2,000 photos of apples and pears as a dataset for sample labeling and model training, we ensure that each fruit category has enough samples to extract features.The training dataset for apple classification is essentially a set of images that we manually labelled with the ground truth and the coordinates of the bounding box positions. We took explicit pictures of apples and pears in a well-lit environment, as shown in the input image, to ensure the quality of the experimental data. The fruits were classified from skin as the criterion for the ripeness classification. We define smooth fruit surface for the maturity, rotten apple is labelled as an overripe class. The purpose of the experiment is to compare and find the best model for fruit detection, we choose epoch as the experimental parameter. It is not enough to pass the complete dataset once in the neural network. We need to pass the complete dataset multiple times. We justify the number of model iterations by adjusting the epoch value to compare the pros and cons of the model. Other parameters are set to batch size as 1, learning rate as 1.00×10^{-4}, betas range (0.90, 0.99), and weight_decay as 5.00×10^{-2}. In our experiments, we make use of PyTorch as our computing platform. We are use of LabelMe (PyTorch) as the tool to make the dataset. In Fig. 1, the orange box is the data set produced by LabelMe.

In fruit ripeness identification, the samples are classified into four groups: True positive (TP), false positive (FP), true negative (TN), and false-negative (FN) based on the combination of their actual classes and the predicted classes. Precision means that the prediction rate of the model is correct in the classes of the samples, as shown in Eq. (11).

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

Table 1 The results of precisions by using YOLOv5x weights

Model	Epoch	Classes	AP50	AP@0.5:0.95
YOLOv5x	30	Ripe apple	0.9995	0.9640
		Overripe apple	0.9995	0.9230
		Ripe pear	0.9995	0.9390
		Overripe pear	0.9995	0.9320
	50	Ripe apple	0.9995	0.9760
		Overripe apple	0.9996	0.9180
		Ripe pear	0.9995	0.9750
		Overripe pear	0.9995	0.9770

In our experimental results, AP50 means that the IOU threshold of the detector is higher than 0.50. AP@50:5:95 refers to the value of IOU taken from 50.00% to 95.00%, the step length is 5.00%, and the average of precisions under these IOUs is calculated.

4 Results

We apply four YOLOv5 weights from small to large. In the process of training the model, we are use of the validation dataset to test the current model in each epoch to obtain the loss and accuracy of the model, verify the loss and accuracy of each epoch. After the model is created and trained, we make use of the test dataset to test the model and get the accuracy. In the test of YOLOv5 model, the larger the weight file of the model, the longer the training time required under the same epoch. YOLOv5 has the same overall architecture for the size (s, m, l, x), but it takes use of different depths and widths in each sub-module, respectively responding to the depth_multiple and width_multiple parameters in the yaml file, where the parameters s, m, l, x of the YOLOv5 model will only be downsampled to 32 times and 3 prediction feature layers are deployed.

We find that too much data will lead to redundancy, the model cannot show good results. In our experiments, we solved the problem of data redundancy. In Tables 1, 2, 3 and 4, we chose the smallest number of epochs to train the model. We see that if the number of model iterations is small, the feature map cannot be transmitted in the deep nets in time, the model cannot be trained enough by using the characteristics of apple images. We also observe that when the number of iterations reaches a balance, the model achieves

Table 2 The results of precisions by using YOLOv5l weights

Model	Epoch	Classes	AP50	AP@0.5:0.95
YOLOv5l	30	Ripe apple	0.9995	0.9550
		Overripe apple	0.9995	0.9580
		Ripe pear	0.9995	0.9370
		Overripe pear	0.9995	0.9320
	50	Ripe apple	0.9994	0.9850
		Overripe apple	0.9996	0.9340
		Ripe pear	0.9994	0.9820
		Overripe pear	0.9994	0.9790

Table 3 The results of precisions by using YOLOv5m weights

Model	Epoch	Classes	AP50	AP@0.5:0.95
YOLOv5m	30	Ripe apple	0.9991	0.9540
		Overripe apple	0.9995	0.9260
		Ripe pear	0.9995	0.9020
		Overripe pear	0.9995	0.8980
	50	Ripe apple	0.9996	0.9370
		Overripe apple	0.9995	0.8960
		Ripe pear	0.9995	0.9440
		Overripe pear	0.9996	0.8620

better results. In Tables 1 and 4, as the IOU threshold increases, we see that the model with a considerable weight produces a minor accuracy difference, hence, model is more stable.

YOLOv5x has considerable weights and parameters, better results have been obtained, not only the precision rate is better, but also the classification accuracy rate is also more than the average. We see our experimental results of Faster R-CNN in the three-class classification experiment; for a specific class, better results are obtained, but for the rest of the two classes, it is relatively poor. The classification results may be affected by the data distribution in the training set.

As the number of epochs increases, the number of weights in the deep nets also grows. The diversity of image data will affect the number of epochs. We need to continuously adjust the number of epochs according to the characteristics of the fruit that the model is trained.

If the weight gradually reduces, the mean average precision rate dips. The rate of one class is slightly lower than that of other classes. So far, we have got the best model by using YOLOv5x. We have solved the problems in our experiments by increasing the number of samples in the dataset and adjusting the weights in the training process to make the cost function drop. We have to consider the weight decay method to adjust YOLOv5 parameters. In view of our current experimental data that has eliminated the environmental noises, we need to consider whether the region of interest (ROI) is able to be segmented with much detail.

In the current experiments, we have improved the results significantly. However, based on the dataset and the results of our experiments, we need a more powerful deep net for further exploration. Pertaining to the YOLO models, we adjust the architecture and delete unnecessary parameters to make the model lighter with the same accuracy

Table 4 The results of precisions by using YOLOv5s weights

Model	Epoch	Classes	AP50	AP@0.5:0.95
YOLOv5s	30	Ripe apple	0.9994	0.8390
		Overripe apple	0.9995	0.9390
		Ripe pear	0.9995	0.8970
		Overripe pear	0.9995	0.8970
	50	Ripe apple	0.9994	0.9060
		Overripe apple	0.9995	0.8800
		Ripe pear	0.9995	0.9230
		Overripe pear	0.9995	0.9220

Table 5 The precision of YOLO using Swin Transformer module

Model	Epoch	AP50	AP@0.5:0.95
Swin + YOLO	10	0.1980	0.6360
	20	0.2050	0.5400
	30	0.5150	0.9950
	50	0.1850	0.5200

in future. But in Table 5, we see that the YOLO model with Swin Transformer model added does not perform well. Although the model keeps the faster training speed of the YOLO model, if the Transformer decoder is used to predict the bounding box of the target object directly, the model has a long convergence time and poor tracking performance.

We see the results of detection transformer in Table 6. The DETR model did not show good results in the recognition of the ripeness of the fruit. We infer that the fruit is a small object. The reason is that no fancy network optimization is needed, such as adding FPN or BiFPN part.

The shortcomings of detection transformer are very obvious. In the test part, the positioning is not accurate. DETR makes up the shortcomings of the anchor-free algorithm. For example, it is useful for processing overlapping scene objects. We infer that the detection transformer model is better for large object detection.

Different from Table 5, Swin Transformer in Table 7 combines Mask R-CNN to exhibit a sequence-to-sequence model feature that makes it easier to combine multimodal data, thereby providing greater flexibility in network architecture design. We see that as the number of iterations increases, the model shows better results.

Table 6 The precision of DETR with ResNet-50

Model	Epoch	Classes	AP@0.5:0.95	Average inference time(seconds)
DETR + ResNet-50	10	Ripe apple	0.7268	0.002
		Overripe apple	0.7374	
		Ripe pear	0.7958	
		Overripe pear	0.7714	
	20	Ripe apple	0.6819	0.004
		Overripe apple	0.7327	
		Ripe pear	0.9663	
		Overripe pear	0.7046	
	30	Ripe apple	0.7953	0.004
		Overripe apple	0.7959	
		Ripe pear	0.6832	
		Overripe pear	0.8560	
	50	Ripe apple	0.7296	0.002
		Overripe apple	0.7764	
		Ripe pear	0.7427	
		Overripe pear	0.7450	

Table 7 The precision of Swin Transformer by using Mask R-CNN

Model	Epoch	AP50	AP@0.5:0.95	Average inference time(seconds)
Swin + Mask R-CNN	10	0.9360	0.8220	0.042
	20	0.9360	0.8360	0.048
	30	0.9580	0.9580	0.042
	50	0.9670	0.9670	0.044

In Tables 6 and 8, we observe that when the accuracy rate is similar, the same anchor-free and backbone model CornerNet requires a longer average inference time. This means that in the application, CornerNet cannot achieve a good detection speed. In Tables 6 and 7, both Swin Transformer and DETR can achieve a faster detection speed. At the same time, Swin Transformer performs better precision than DETR with the similar computing speed.

5 Conclusion

In this paper, we have implemented fruit classification with the classes of multilevel ripeness. Our models are able to automatically classify the ripeness degree of fruits. The experimental results also show that the number of epochs needs to be controlled. Too many iterations will lead to model redundancy, while too few samples make the model underfitting, ultimately unable to get the natural characteristics of the apple images, and thus unable to get satisfactory results.

Our contribution from this paper is to find the best model for fruit maturity recognition. The model is able to be applied to automated multiple fruits classification in the agricultural industry. We attained the optimal model by adjusting the model and parameters. At the same time, we also analyzed the reasons that affect our experiments, which are helpful for our subsequent experiments. From the detection results, the YOLO model has better results in detection and is more stable. The Swin Transformer model and Mask RCNN can also achieve fast and stable detection. However, the addition of the transformer mechanism to the YOLO model did not yield better results. The mask module set by the attention mechanism of Swin Transformer can better accomplish the shift of the feature map in the window.

In the future, we will have multistage goals in our experiment. High-accuracy fruit detection in unstructured orchard environments remains particularly challenging due to varying lighting conditions and occlusions from shadows that affect experimental accuracy.

Table 8 The precision of CornerNet

Model	Epoch	AP@0.5:0.95	Average inference time(seconds)
CornerNet + Hourglass	10	0.718	0.60
	20	0.245	0.58
	50	0.789	0.56

In this paper, we optimized the fruit dataset, but in the complex background, environmental noises, dynamic detection and tracking of fruit targets from different perspectives, and the accuracy of identifying occluded visual targets with a fixed perspective, the recognition process, positioning, the impact of these issues on the experiments are also research topics for our future projects.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Data Availability The data is available upon request.

Declarations

Conflict of interest This work has not any funding support, it has not any conflicts of interests or competing interests.

References

1. Abozeid A, Alanazi R, Elhadad A, Taloba AI, Abd El-Aziz RM. (2022). A large-scale dataset and deep learning model for detecting and counting olive trees in satellite imagery. Computational Intelligence and Neuroscience
2. Agushinta RD, Medyawati H, Jatnika I. (2017). A method of cloud and image-based tracking for Indonesia fruit recognition. IEEE Int Conf Eng Tech Soc Sci (ICETSS). pp. 1–5
3. Ahmad T, Ma Y, Yahya M, Ahmad B, Nazir S (2020) Object detection through modified YOLO neural network. Scientific Programming 2020:1–10. https://doi.org/10.1155/2020/8403262
4. Alkalouti H, Masre M. (2021) Encoder-decoder model for automatic video captioning using YOLO algorithm. IEEE Int IOT, Electron Mechatron Conf (IEMTRONICS). pp. 1–4
5. Arivazhagan S, Shebiah RN, Nidhyanandhan SS, Ganesan L (2010) Fruit recognition using color and texture features. J Emerg Trends Comput Inform Sci 1(2):90–94
6. Arkin E, Yadikar N, Muhtar Y, Ubul K. (2021). A survey of object detection based on CNN and Transformer. IEEE International Conference on Pattern Recognition and Machine Learning (PRML), pp. 99–108
7. Bochkovskiy A, Wang CY, Liao HYM. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934
8. Chen Q, Wang Y, Yang T, Zhang X, Cheng J, Sun, J. (2021). You only look one-level feature. IEEE/CVF ConfComput Vis Patt Recog. pp. 13039–13048
9. Choudhury A, Biswas A, Prateek M, Chakrabarti A (2021) Agricultural informatics: Automation using the IoT and machine learning. John Wiley & Sons, Incorporated
10. Dai Z, Cai B, Lin Y, Chen J. (2021). Up-DETR: Unsupervised pre-training for object detection with transformers. IEEE/CVF Conf Comp Vis PattRecog. pp. 1601–1610
11. Dou Q, Yan M (2021) Ocean small target detection in SAR image based on YOLOv5. J Eng 7(3):167–173
12. Fu Y, Nguyen M, Yan WQ (2022) Grading methods for fruit freshness based on deep learning. SN Comput Sci 3(4):264
13. Glorot X, Bordes A, Bengio Y.(2011) Domain adaptation for large-scale sentiment classification: A deep learning approach. ICML
14. Han X, Dang Y, Mei L, Wang Y, Li S, Zhou X. (2019) A novel part of speech tagging framework for NLP-based business process management. IEEE International Conference on Web Services (ICWS) pp. 383–387
15. Hendria WF, Phan QT, Adzaka F, Jeong C. (2021) Combining Transformer and CNN for object detection in UAV imagery. ICT Express

16. Jiménez AR, Jain AK, Ceres R, Pons JL (1999) Automatic fruit recognition: A survey and new results using range/attenuation images. Pattern Recogn 32(10):1719–1736
17. Kabir MS, Ndukwe IK, Awan EZS. (2021) Deep learning inspired vision based frameworks for drone detection. IEEE International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pp. 1–5
18. Kim S, Kim H (2021) Zero-centered fixed-point quantization with iterative retraining for deep convolutional neural network-based object detectors. IEEE Access 9:20828–20839
19. Kousik N, Natarajan Y, Raja RA, Kallam S, Patan R, Gandomi AH. (2021) Improved salient object detection using hybrid convolution recurrent neural network. Expert Syst Appl. 166, 114064
20. Kuznetsova A, Maleva T, Soloviev V. (2020) Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images. Int Symp Neural Netw. pp. 233–243
21. Mauri A, Khemmar R, Decoux B, Ragot N, Rossi R, Trabelsi R, Savatier X (2020) Deep learning for real-time 3D multi-object detection, localization, and tracking: Application to smart mobility. Sensors 20(2):532
22. Mekhalfi M, Nicolò C, Bazi Y, Al Rahhal M, Al Sharif N, Al Maghayreh E. (2021) Contrasting YOLOv5, Transformer, and EfficientDet detectors for crop circle detection in desert. IEEE Geosci Remote Sens Lett
23. Mhalla A, Chateau T, Amara NEB (2019) Spatio-temporal object detection by deep learning: Video-interlacing to improve multi-object tracking. Image Vis Comput 88:120–131
24. Pal SK, Pramanik A, Maiti J, Mitra P. (2021). Deep learning in multi-object detection and tracking: State of the art. Appl Intell. pp.1–30
25. Qi J, Nguyen M, Yan W (2022). Waste classification from digital images using ConvNeXt. In Pacific-Rim Symp Image Vid Technol
26. Redmon J, Divvala S, Girshick R, Farhadi A. (2016). You Only Look Once: Unified, real-time object detection. IEEE Conf Comput Vis Patt Recog. pp. 779–788
27. Tang Z, Yang J, Pei Z, Song X. (2021). Coordinate-based anchor-free module for object detection. Appl Intell. 1–15
28. Tang Z, Yang J, Pei Z, Song X (2021) Coordinate-based anchor-free module for object detection. Appl Intell 22:1–5
29. Tang Y, Zhang Y, Zhu Y. (2020) A research on the fruit recognition algorithm based on the multi-feature fusion. International Conference on Mechanical, Control and Computer Engineering (ICMCCE) pp. 1865–1869
30. Tsai M, Tseng H (2021) Enhancing the identification accuracy of deep learning object detection using natural language processing. J Supercomput 2:1–6
31. Ünal HB, Vural E, Savaş BK, Becerikli Y. (2020). Fruit recognition and classification with deep learning support on embedded system (FruitNet). Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1–5
32. Wang X, Hua X, Xiao F, Li Y, Hu X, Sun P (2018) Multi-object detection in traffic scenes based on improved SSD. Electronics 7(11):302
33. Wu D, Lv S, Jiang M, Song H (2020) Using channel pruning-based YOLOv4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Comput Electron Agric 178:105742
34. Xia Y, Nguyen M, Yan WQ. (2022). A real-time kiwifruit detection based on improved YOLOv7. In IVCNZ, pp. 48–61
35. Xiao B. (2019) Apple Ripeness Identification Using Deep Learning. Rearch Report, Auckland. University of Technology, New Zealand
36. Xiao B, Nguyen M, Yan W. (2021). Apple ripeness identification using deep learning. In: Int Symp Geom Vis (ISGV). pp.53–67
37. Zhang X, Wan F, Liu C, Ji X, Ye Q. (2021) Learning to match anchors for visual object detection. IEEE Trans Patt Anal Mach Intell (2021)
38. Zhao K, Yan WQ. (2021) Fruit detection from digital images using CenterNet. In Int Symp Geomet Vis (ISGV)