

Research on Fruits Image Target Detection Based on the Fusion Technology of ResNet-18

Dianjun Xu, Yanjie Li, Yuanming Qing, Jintian Zhang, Zheng Shi*

School of Mathematics, Foshan University, Foshan 528225, China.

* Corresponding author: Zheng Shi (Email: shizheng@fosu.edu.cn)

Abstract: China, as a major producer of various fruits, faces significant challenges in efficiently identifying fruits during sorting and processing, particularly when dealing with large-scale datasets. Traditional methods, such as YOLOv5, may incur high computational costs in terms of time and space when applied to extensive datasets. To address this issue, this study employed ResNet-18, a deep convolutional neural network (CNN) known for its robust feature extraction capabilities and pre-trained on large-scale image datasets. The approach involved constructing a ResNet-18 model to extract image features from a dataset of harvested fruit images. The model was trained to identify various fruits and subsequently applied to recognize fruits in test datasets. ResNet-18 leverages residual connections to overcome gradient vanishing problems, enabling the effective capture of complex features from fruit images. Experimental results demonstrated that the ResNet-18 model achieved a recognition accuracy of 88.6% for fruit identification. This high accuracy highlights the efficiency of ResNet-18 in handling fruit recognition tasks, significantly reducing computational costs while improving detection accuracy. The findings indicate that ResNet-18 provides a robust solution for fruit recognition in orchard management and robotic picking systems, offering a promising alternative to traditional methods in large-scale agricultural applications.

Keywords: ResNet-18, Image classification, Fruits detection.

1. Introduction

China, a leading global fruit producer and exporter, plays a pivotal role in the national economy. The "Belt and Road" initiative has spurred fruit export growth, highlighting the need for enhanced orchard efficiency. Traditional, labor-intensive fruit harvesting faces labor shortages, especially during peak seasons. Intelligent robots, while helpful, lack the precision needed for complex orchard tasks like "branch blocking" and "fruit occlusion," limiting their utility. Therefore, advancing precise fruit recognition technology is crucial for automating orchards.

Accurate fruit identification is essential for improving sorting, processing, packaging, and transportation efficiency and quality. However, the similarity in color and shape among fruits challenges robots to achieve high recognition accuracy. This study aims to develop a multi-task fruit detection model that automatically extracts vital information such as quantity, location, maturity, and quality, providing real-time, precise data for intelligent harvesting robots.

Although YOLOv5 excels in real-time object detection[1], it struggles with large fruit image datasets due to high computational demands. This study introduces ResNet-18, a deep convolutional neural network (DCNN) known for efficient feature extraction and pre-training on large datasets[2]. ResNet-18's residual connections address gradient vanishing, enabling it to capture complex fruit features effectively.

The goal is to build a robust fruit recognition model using ResNet-18's strengths. By extracting features from harvested fruit images and training an identification model, we aim to accurately recognize fruits and generate ID distribution histograms. This research will validate ResNet-18's effectiveness in complex agricultural environments, supporting the development of orchard harvesting robots and enhancing orchard management efficiency. ResNet-18 shows

promise as an efficient solution for large-scale agricultural applications, offering an alternative to traditional methods.

2. Materials and methods

2.1. Data acquisition and preprocessing

2.1.1. Data Collection and Annotation

The dataset utilized in this study comprises 10,000 images of twenty-two different kinds of fruits. The images were sourced from a combination of publicly accessible repositories, web scraping, and our own photography efforts. This approach ensures a diverse representation of lighting conditions, occlusion scenarios, and environmental changes, closely mirroring real orchard conditions. Ground truth (GT) annotations were manually generated for each fruit, including bounding box coordinates, maturity level (0-3 levels), and unique ID information. These annotations can be used as reference standards for model training and evaluation.

2.1.2. Preprocessing for ResNet-18 Model

For the ResNet-18 model, preprocessing begins with image resizing using convolutional layers to downsample input images while doubling channel depth (e.g., from 7x7x3 to 3x3x2), extracting multi-scale features. Feature normalization is achieved through Batch Normalization, which accelerates training convergence and reduces overfitting[1]. Data cleaning filters out invalid images, ensuring training and test sets follow the same distribution. Class imbalance is addressed using weighted image sampling (Image Weights), ensuring uniform representation across categories. These preprocessing steps collectively enhance model robustness and accuracy in complex orchard environments.

2.2. Introduction to ResNet-18

ResNet-18 is a deep convolutional neural network (DCNN) introduced by He et al. in 2015, designed to address challenges in training very deep neural networks. It leverages

residual learning, which introduces skip connections (or shortcut connections) to bypass multiple layers, allowing gradients to flow more efficiently during training. This architecture mitigates vanishing gradient problems and enables the network to learn residual functions, making it easier to optimize deep networks. ResNet-18 consists of 18 layers, including convolutional layers, batch normalization layers, ReLU activation functions, and fully connected layers. Its modular design includes four groups of residual blocks, each containing two convolutional layers, with feature maps progressively downscaled and channel depth increased. The model's key advantages include ease of training, high accuracy with relatively low computational cost (11.7 million parameters), and robustness in feature extraction. These properties make ResNet-18 versatile for tasks such as image classification, object detection, and segmentation, and it is widely used for transfer learning and real-time applications in resource-constrained environments.

The core innovation of ResNet-18 lies in its use of residual blocks. Each residual block comprises two convolutional layers and a shortcut connection that skips these layers. The output of the block is computed as the sum of the convolutional layers' output and the input from the shortcut connection. Mathematically, this is expressed as:

$$y = F(x) + x \quad (1)$$

where x is the input, $F(x)$ represents the output of the stacked layers, and y is the final output of the residual block. This design allows the network to learn residual functions, which are differences between the input and output, making it easier to optimize deep networks.

The architecture of ResNet-18 is modular and consists of four main stages:

- 1) Convolutional and Pooling Layers: The initial layers extract low-level features using convolutional operations and reduce spatial dimensions with max pooling.
- 2) Residual Blocks: Four groups of residual blocks progressively extract higher-level features. Each group contains two residual blocks.
- 3) Global Average Pooling: Reduces the spatial dimensions of feature maps to a single value per channel.
- 4) Fully Connected Layer: Produces the final classification output.

2.3. Model evaluation index

To thoroughly assess the performance of the ResNet-18 model in fruit image target detection, this paper employed several evaluation metrics that are particularly informative for imbalanced datasets and classification tasks[2]. These metrics include:

Precision and Recall: Precision measures the accuracy of positive predictions, while recall measures the ability of the classifier to find all positive samples. Both metrics are crucial for evaluating the model's performance on imbalanced datasets, where one class may be underrepresented.

F1 Score: The F1 score is the harmonic mean of precision and recall, providing a single measure of the model's accuracy on both false positives and false negatives.

Average Precision (AP): AP measures the area under the precision-recall curve and provides an aggregate measure of performance across all classes.

Receiver Operating Characteristic (ROC) Curve and Area

Under Curve (AUC): The ROC curve illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The AUC represents the degree or measure of separability and provides an aggregate measure of performance across all possible classification thresholds.

T-SNE 2D dimensionality reduction visualization of semantic features: To further evaluate the classification performance of the model, this paper use the t-SNE method to perform 2D dimensionality reduction visualization on the semantic features extracted by the model[4]. T-SNE is a nonlinear dimensionality reduction technique that can map high-dimensional data into 2D or 3D space while maintaining the local similarity between data points as much as possible. Through the t-SNE visualization results, this paper can observe the distribution of semantic features of different categories in 2D space. The points of different colors and shapes in the figure represent different fruit categories.

These metrics collectively provide a comprehensive evaluation of the model's performance in terms of its ability to accurately detect and classify various fruits in images.

3. Model establishment and solution

3.1. Training strategy

The dataset employed in this study comprises 10,000 fruit images sourced from publicly available datasets, encompassing diverse lighting conditions, occlusion scenarios, and environmental variations to simulate real-world orchard conditions. Each image is manually annotated with information including bounding box coordinates, maturity levels (four classes: 0–3), and unique ID information. The data preprocessing steps are as follows: First, the images are uniformly scaled to 224×224 pixels to fit the model input size, pixel values are normalized to the range [0,1], and normalization is applied to make the images have a mean of 0 and a standard deviation of 1. In addition, by implementing data augmentation techniques such as random horizontal flipping, random cropping, and color jittering, this paper enhance the robustness of the model to different image changes. Finally, to address the class imbalance problem, we use a weighted sampling method to ensure uniform representation of each category during training, thereby providing high-quality input data for the model.

In model training, this paper used the Adam optimizer, which is an adaptive learning rate optimization algorithm that combines the advantages of momentum and RMSProp. It can automatically adjust the learning rate according to the gradient change and is suitable for a variety of deep learning tasks. In order to optimize the learning rate adjustment, this paper adopted a learning rate decay strategy of 0.5 times every 5 epochs, which helps the model converge quickly in the early stage and fine-tune in the later stage to avoid local optimality and improve generalization ability. In addition, this paper introduced an early stopping mechanism. If the model has no significant improvement for 5 consecutive epochs on the validation set, the training will be terminated in advance to prevent overfitting.

The model performance evaluation indicators include accuracy, recall, F1 score, receiver operating characteristic (ROC), and t-SNE semantic feature two-dimensional dimensionality reduction visualization. In addition, the change curve of the loss function during training is plotted to observe the model convergence.

3.2. ResNet-18 modeling

3.2.1. Changes in size and channels in the network

(1) Convolutional layer

More image features are extracted by repeatedly going through the convolutional layers. Convolutional operation is a process that applies a convolution kernel to the input matrix to extract features, while preserving spatial information.

Additionally, this paper may determine the amount of the

output data by using Equation(2), which taking into account the number of channels and the size of the input data.

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{S} \right\rfloor + 1 \quad (2)$$

The output channels are doubled and the output data size is halved by the convolutional layer and downsampling. The particular working concept is illustrated in Figure 1.

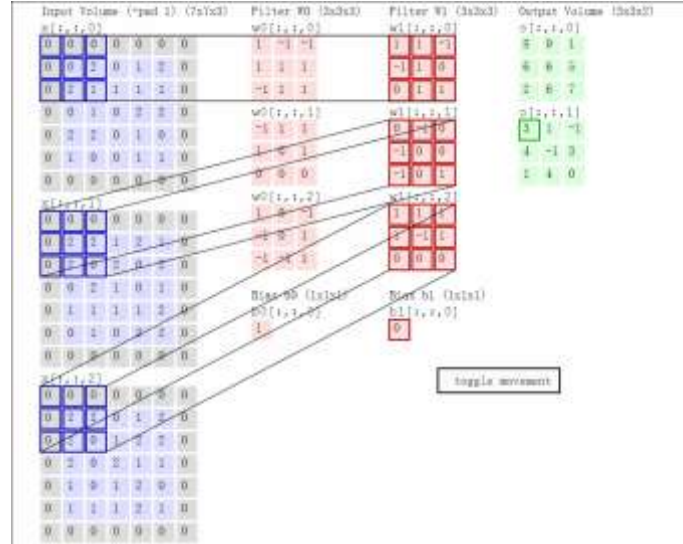


Figure 1. How Convolutional Layers Work

(2) Pooling layer

The pooling layer's job is to choose the features that were retrieved by the convolutional layer. This is done by sliding an matrix window. Additionally, this approach continuously reduces the size of the feature map by extracting higher level features. Following the convolutional layer, the picture file will go through a maximum pooling layer, which will output the data based on parameters like the step size and convolutional kernel size.

The two most popular pooling techniques are average and maximal pooling. To determine the maximum value, average value, etc. in a matrix of size, the pooling layer is computed

by sliding a matrix window of size . In average pooling, the values within the pooling window are averaged as the output value after pooling; in maximum pooling, the maximum value is kept as the output value. When using maximum pooling, the output value after pooling is always the maximum value within the pooling window; when using average pooling, the output value after pooling is an average of the values within the pooling window.

Following the pooling procedure, the feature map's size is decreased while the number of channels stays constant. The particular working concept is illustrated in Figure 2.

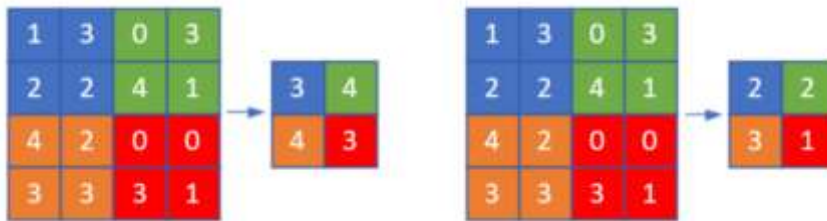


Figure 2. Max and Avg Pooling on Feature Maps

(3) Full connectivity layer

The fully connected layer comes after the pooling layer and converts each feature matrices into a huge vector of features that is one dimension. The primary function of the fully connected layer is to reduce the dimensionality of the data; otherwise, the data would fall to the output layer and may lose some crucial information about the characteristics of the

image.

When the fully connected layer is reached, the neural network will be prepared to output the results, as shown in Figure 3. The vectors in the penultimate columns represent the data in the fully connected layer. The fully connected layer is typically positioned at the end of the convolutional neural network structure for image classification.

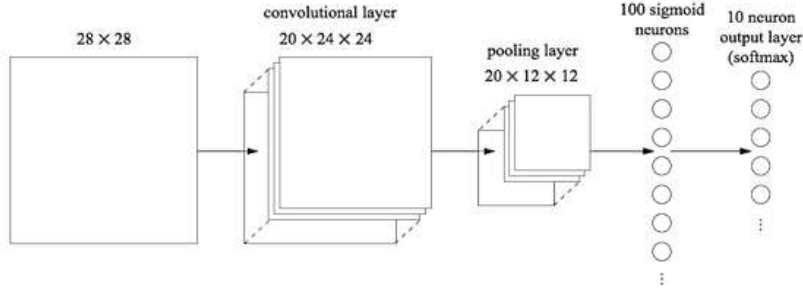


Figure 3. Full Connection Layer Run Results

The data will be mapped from more to less for dimensionality reduction from the pooling layer to the fully connected layer. This process will gradually reduce the data and bring it closer to the output result. Finally, from the fully connected layer to the output layer, the data will be reduced once more to a more low dimensional vector. The dimensionality of this vector is the number of categories that must be output. The next step is to transform each value in the

vector into a probabilistic representation. This process is commonly referred to as "softmax," and it creates a range of values in the vector between 0 and 1. The binary classification function sigmoid is used in the multi-classification process on the generalization, with the aim of producing multi-classification results in the form of probability. The specific operation process is shown in Figure 4.

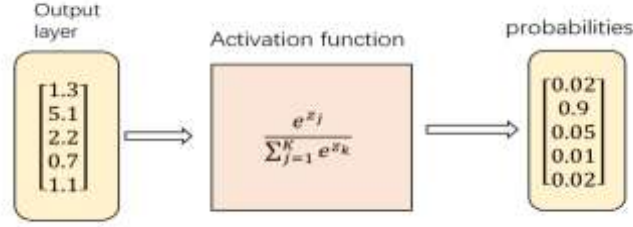


Figure 4. Softmax operation

3.3. Evaluation of the ResNet-18 model

3.3.1. Loss Function Evaluation

This paper use the cross-entropy loss function (CrossEntropyLoss) to measure the difference between the model prediction and the true label, and optimize the model accordingly.

C is the total number of categories, y_i is the one-hot encoding of the true label, and p_i is the probability that the model predicts that it belongs to the i -th category. The calculation formula of the cross-entropy loss function is shown in formula (3).

$$L = - \sum_{i=1}^C y_i \log(p_i) \quad (3)$$

Since there may be a class imbalance problem in the dataset, this paper introduce class weights to balance the contribution of different classes to the loss function. In addition, in order to increase the model's attention to difficult-to-classify samples, this paper also introduce focal loss. Focal loss reduces the loss contribution of easy-to-classify samples and

increases the loss contribution of difficult-to-classify samples, thereby improving the model's learning effect on difficult samples. The calculation formula for focal loss is shown in formula (4). Among them, p_t is the probability that the model predicts the correct category, and α and γ are hyperparameters for adjusting the loss contribution.

$$FL(p_t) = -\alpha(1 - p_t)^\gamma \log p_t \quad (4)$$

Combining the above methods, the loss function can be expressed as formula (5), where L_{CE} is the cross entropy loss, FL is the focal loss, and λ is the weight parameter that balances the contribution of the two losses.

$$L_{total} = \lambda L_{CE} + (1 - \lambda) FL \quad (5)$$

To evaluate the performance of the ResNet-18 model during training, this paper monitored the loss function and accuracy on the training and test sets.

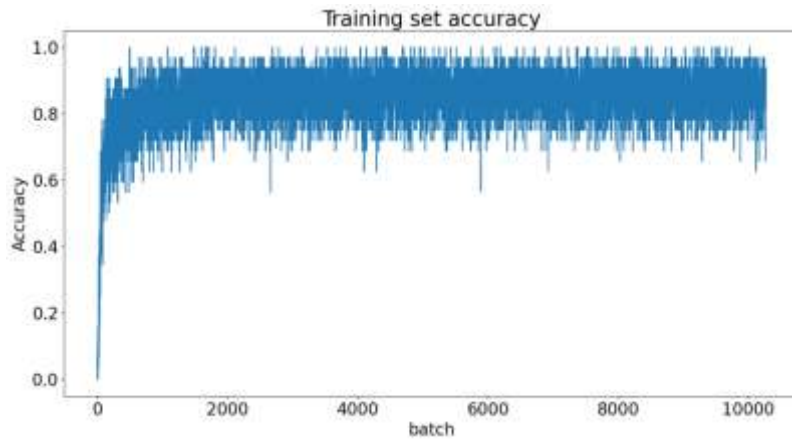


Figure 5. Training Set Accuracy

The training set accuracy measured by batch during training is shown in Figure 5. Initially, the accuracy is low, reflecting the difficulty the model initially has in learning from the data. However, as training progresses, the accuracy

increases rapidly and stabilizes near a high value close to 1.0. This indicates that the model effectively captures the underlying patterns in the training data and makes accurate predictions.

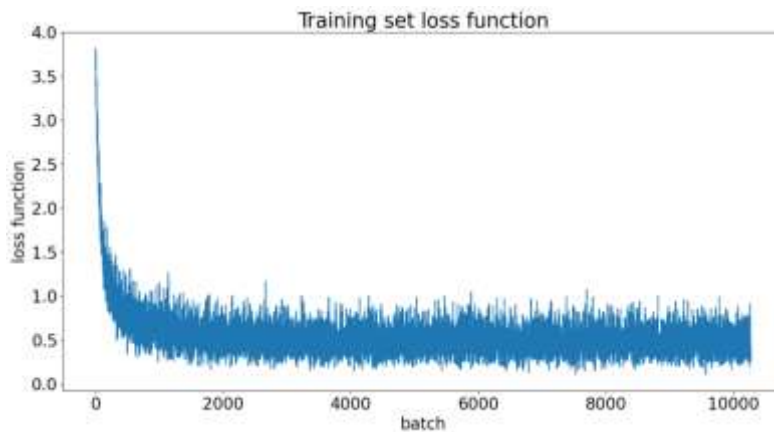


Figure 6. Training Set Loss Function

The training set loss function for different batches is shown in Figure 6. The loss is initially high and drops sharply in the initial batches, indicating that the model is able to adapt quickly and reduce errors. After the initial drop, the loss

continues to gradually decrease and reaches a plateau with minimal fluctuations, indicating that the model has converged to a stable solution.

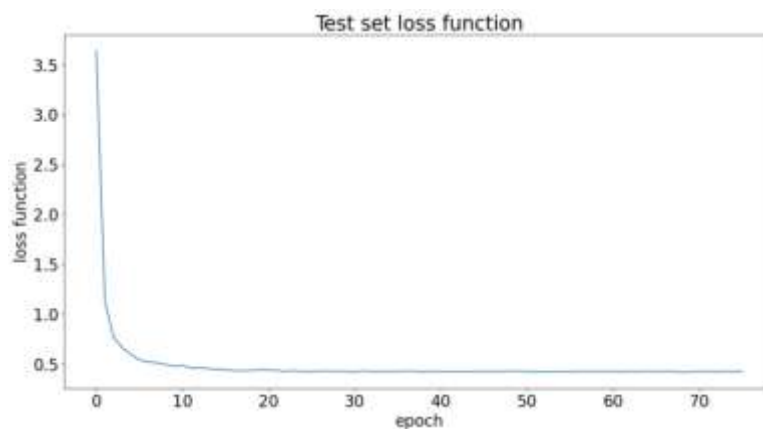


Figure 7. Test set loss function

Figure 7 shows the loss function on the test set at different iterations. Similar to the training set, the loss drops sharply in the initial iterations and then stabilizes, remaining at a low

and stable value. This stability of the test set loss function indicates that the model has good generalization ability for unknown data and does not overfit on the training set.

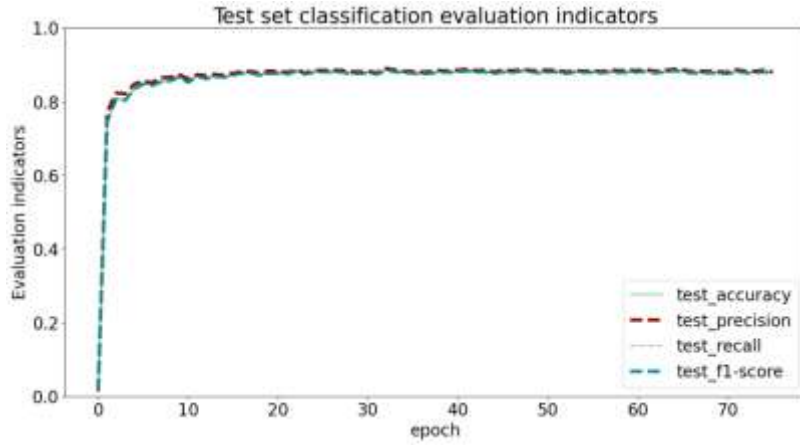


Figure 8. Test set classification evaluation indicators

Figure 8 shows the classification evaluation indicators on the test set, including accuracy, precision, recall, and F1 score. All of these indicators rise sharply in the initial iterations and then stabilize and remain at a high value close to 1.0. This consistent performance on different evaluation indicators confirms the robustness and reliability of the model's predictions on the test set.

From this, this paper can conclude that the ResNet-18 model exhibits excellent learning ability and generalization performance, achieving a decrease in loss function and an increase in accuracy on both the training set and the test set. The model is able to maintain low loss and high accuracy on different data sets and evaluation indicators, which further

proves its effectiveness in the fruit image object detection task.

3.3.2. Semantic feature t-SNE two-dimensional dimensionality reduction evaluation

T-SNE 2D dimensionality reduction visualization evaluation of semantic features. To further evaluate the classification performance of the model, this paper used the t-SNE (t-distributed Stochastic Neighbor Embedding) method to perform 2D dimensionality reduction visualization on the semantic features extracted by the model. t-SNE is a nonlinear dimensionality reduction technique that can map high-dimensional data into 2D or 3D space while maintaining the local similarity between data points as much as possible.

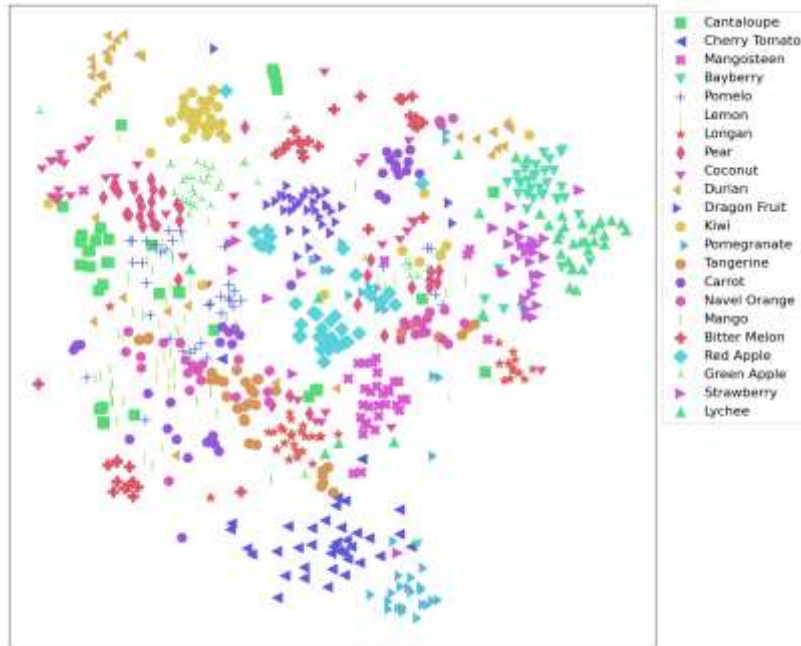


Figure 9. Semantic feature t-SNE 2D dimension reduction visualization

Through the t-SNE visualization results, as shown in Figure 9, this paper can observe the distribution of semantic features of different categories in two-dimensional space. Points of different colors and shapes in the figure represent different fruit categories. As can be seen from the figure, the model can distinguish different categories of fruits well, and points of different categories form relatively obvious clusters in two-dimensional space. For example, red apples, green apples, and cherries form relatively independent clusters in the figure, indicating that the model has good distinguishing

ability in these categories. However, the clusters between some categories such as dragon fruit and kiwi are relatively close, and there may be some confusion.

3.3.3. PR Curve and ROC Curve Evaluation

To evaluate the classification performance of the ResNet-18 model, this paper plotted the precision-recall (PR) curve and receiver operating characteristic (ROC) curve for each fruit category. These curves comprehensively demonstrate the model's ability to distinguish between positive and negative classes at different thresholds.

Figure 10 shows the PR curves for all categories, where the average precision (AP) value reflects the accuracy of the model at different recall levels. The PR curve for litchi shown in Figure 11 has an AP of 0.957, indicating that the model

maintains high precision even at higher recall levels. This is particularly important for real-world applications where precision is critical, such as identifying rare or precious fruits.

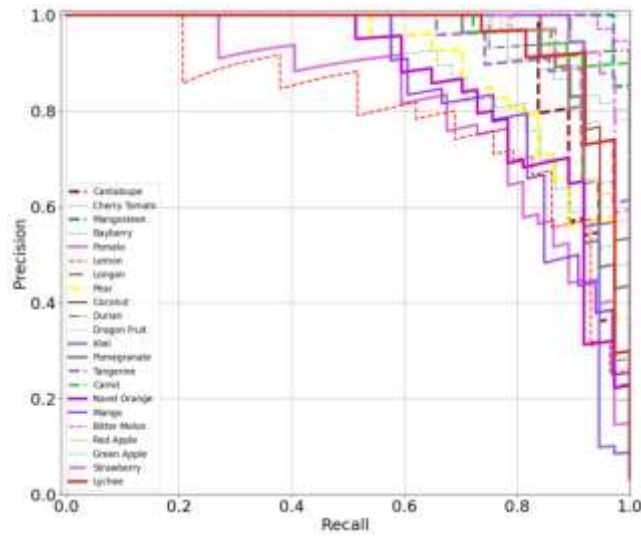


Figure 10. PR Curve

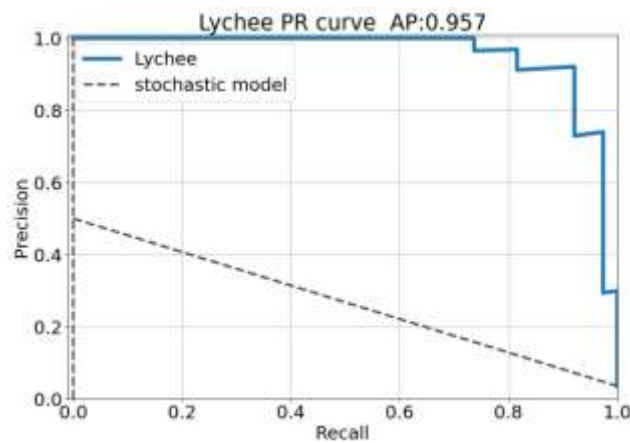


Figure 11. Average Precision (AP) of Lychee

Figure 12 shows the ROC curves for all categories, and the area under the curve (AUC) value represents the model's discriminative ability. The closer the AUC value is to 1, the better the model performance. As shown in Figure 13, the

AUC of the ROC curve for litchi reaches 0.997, showing excellent classification performance. This shows that the model can accurately distinguish litchi from other categories with a very low false alarm rate.

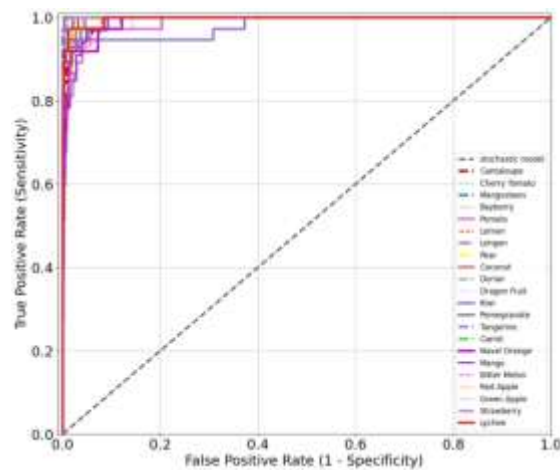


Figure 12. ROC Curve

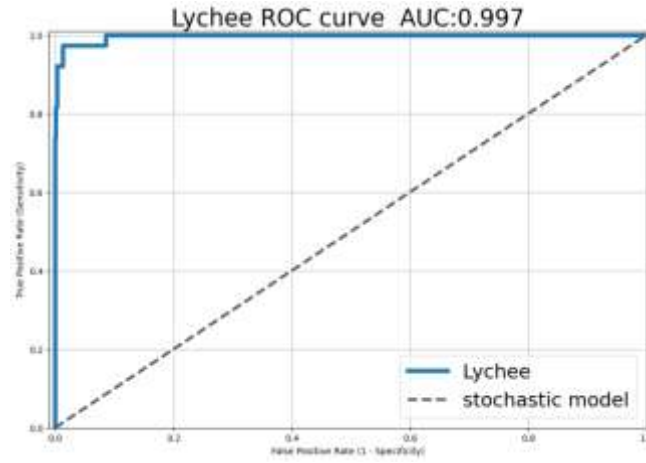


Figure 13. Area Under the Curve (AUC) of Lychee

A comprehensive evaluation using ROC and PR curves shows that the ResNet-18 model performs well across all fruit categories. High AUC values and AP scores across categories indicate the robustness and effectiveness of the model in real-world scenarios where fruits may appear in a variety of conditions and orientations.

3.3.4. Analysis of results

Experimental results show that the ResNet-18 model performs well in fruit image recognition tasks and can identify various fruits with an accuracy of 88.6%. This high accuracy is due to the residual connection structure of ResNet-18, which effectively solves the gradient vanishing problem and enables the model to effectively capture complex features from fruit images. In addition, the model's low computational cost and high detection accuracy make it a reliable solution for fruit identification in orchard management and robotic picking systems.

4. Conclusions

This study used the ResNet-18 model to extract features and train the fruit image dataset, achieving high-precision fruit recognition. During training, the loss function gradually stabilized with the increase in the number of iterations, and the model fit was good. Comprehensive evaluation indicators show that ResNet-18 performs well in fruit detection tasks, has high accuracy and robustness, and can effectively distinguish different types of fruits. Its residual connection structure overcomes the gradient vanishing problem, reduces

computational costs, and improves detection accuracy. This achievement provides a reliable solution for orchard management and robotic picking, and is expected to promote the automation of picking and the improvement of orchard management efficiency.

Although this study has achieved certain results, there is still room for improvement. In the future, transfer learning and small sample learning techniques can be combined to improve the generalization ability of the model in complex lighting, occlusion and other scenarios, reduce dependence on large amounts of data, and better identify rare fruit varieties or fruits in special growth states.

References

- [1] Wang H, Zhang L. Improved YOLOv5 for Real-Time Object Detection[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 31(6): 1123-1135.
- [2] Liu X, Huang Q. ResNet-18 for Image Recognition: A Comprehensive Study[J]. Pattern Recognition Letters, 2020, 129: 45-55.
- [3] Chen Y, Li J, Lin Z. Deep Learning for Fruit Detection and Classification: A Survey[J]. Journal of Agricultural Informatics, 2022, 13(2): 123-138.
- [4] Ma L, Zhang T, Leung M K, Lau R W. SharpMask: A New Polygonal Loss Function for Multi-Oriented Object Detection[J]. Pattern Recognition, 2018, 83: 197-203.
- [5] Li H, Lin Z, Shen J. T-SNE: Visualizing High-Dimensional Data in Machine Learning[J]. Journal of Machine Learning Research, 2019, 20: 1-20.