Postprint: A Lightweight Modified YOLOv5 Method for Apple Tree Yield Estimation

Authors: Li Zhijun, Yang Shenghui, Shi Deshuai, Liu Xingxing, Zheng Yongjun

Date: 2023-02-17T00:00:00+00:00

Abstract

Fruit tree yield estimation is an important component of orchard management. To improve the accuracy of in-situ yield estimation in apple orchards, this study proposes a yield measurement method comprising an improved YOLOv5 fruit detection algorithm and a yield fitting network. UAV and Raspberry Pi cameras were utilized to collect in-situ images of apple orchards at different coloring times after bag removal, forming a sample dataset. The YOLOv5 algorithm was improved by replacing standard convolutions with depthwise separable convolutions and adding attention mechanism modules, which addresses the issues of lack of attention preference during feature extraction and parameter redundancy in the network, thereby enhancing detection accuracy and reducing the computational burden from network parameters. Images were input to obtain estimated fruit counts and the total area of bounding boxes. Using the aforementioned detection results as input and actual yield as output, the yield fitting network was trained to obtain the final yield estimation model. Yield estimation experimental results demonstrate that the improved YO LOv5 fruit detection algorithm can enhance recognition accuracy while improving lightweight characteristics. Compared with the pre-improvement version, detection speed can be increased by up to 15.37%, and the average mAP can reach 96.79%. Test results on different datasets indicate that lighting conditions, coloring time, and whether the background includes a white cloth all exert certain influences on algorithm accuracy. The yield fitting network can predict fruit tree yield with reasonable accuracy, achieving coefficients of determination R^2 of 0.7967 and 0.7982 on the training and test sets, respectively, and root mean square errors RMSE of 1.5317 kg and 1.4021 kg, respectively; the prediction accuracy remains basically stable across samples with different yields. The fruit tree yield estimation model achieves relative error ranges within 7% and 13% under conditions with and without a white cloth background, respectively. The fruit tree yield estimation method based on lightweight improved YOLOv5 proposed in this

study exhibits good accuracy and effectiveness, basically satisfying the requirements for on-tree apple yield estimation in natural environments, and provides a technical reference for intelligent agricultural equipment in modern orchard settings.

Full Text

Yield Estimation Method of Apple Tree Based on Improved Lightweight YOLOv5

LI Zhijun[1,2], YANG Shenghui[1,2], SHI Deshuai[1,2], LIU Xingxing[1,2], ZHENG Yongjun[1,2*]

[1]College of Engineering, China Agricultural University, Beijing 100083, China
[2]Yantai Institute of China Agricultural University, Yantai 264670, China

Abstract: Fruit tree yield estimation is a critical component of orchard management. To improve the accuracy of in-situ yield estimation for apple orchards, this study proposes a yield measurement method that integrates an improved YOLOv5 fruit detection algorithm with a yield fitting network. In-situ images of bag-removed apples at different coloring stages were collected using an unmanned aerial vehicle and Raspberry Pi camera to form a sample dataset. The algorithm was enhanced by replacing standard convolutions with depthwise separable convolutions and adding an attention mechanism module to address the issues of feature extraction without attention preference and parameter redundancy, thereby improving detection accuracy while reducing computational burden. Using images as input, the method estimates fruit quantity and total bounding box area. These detection results serve as inputs to train the yield fitting network, with actual yield as output, yielding the final production estimation model. Experimental results demonstrate that the improved fruit detection algorithm enhances both lightweight characteristics and recognition accuracy. Compared with the original version, detection speed improved by up to 15.37%, with mean average precision (mAP) reaching 96.79%. Tests under different datasets revealed that lighting conditions, coloring time, and background presence of white cloth all affect algorithm accuracy. The yield fitting network effectively predicted tree yields, achieving coefficients of determination (R^2) of 0.7967 and 0.7982 on training and test sets, respectively. Prediction accuracy remained stable across different yield samples, with root mean square errors (RMSE) of 1.5317 kg and 1.4021 kg, respectively. Tests under varying conditions showed the fruit tree yield model achieved relative error ranges within 7% with white cloth background and within 13% without white cloth background. The proposed lightweight improved YOLOv5-based yield estimation method demonstrates good precision and effectiveness, meeting the requirements for on-tree apple yield measurement in natural environments and providing technical references for intelligent agricultural equipment in modern orchards.

Keywords: apple in-situ yield estimation; deep learning; fruit detection; BP

neural network; YOLOv5

Fruit tree yield estimation not only helps fruit growers understand tree growth status and estimate overall orchard output value but also provides quantitative basis for rational harvest arrangement. Traditional yield estimation methods rely primarily on manual visual counting, which demands high experience from estimators and suffers from intensive labor and low accuracy. To automate apple yield measurement, researchers have begun utilizing machine vision technologies, focusing mainly on extracting fruit count information from tree images, while yield estimation based on image information requires further investigation.

Cheng et al. proposed using fruit area ratio, fruit count ratio, small fruit area ratio, and fruit-to-leaf ratio as features input to a neural network for fitting fruit tree yield. Crtomir et al. collected image data of "Golden Delicious" and "Braeburn" apple trees from post-bagging to harvest, using fruit count as input and yield as output to construct an artificial neural network for model training and testing. This method requires multi-period data collection and is only suitable for yield estimation of apple trees approaching or at maturity. Roy et al. developed a semi-supervised clustering method for apple recognition based on color identification and an unsupervised clustering method using spatial attributes to estimate counts from apple clusters with arbitrary complex geometries, integrating them into a complete end-to-end computer vision system that uses images captured by a single camera as input to predict orchard yield, achieving accuracies of 91.98%~94.81% across different datasets.

Deep learning-based object detection algorithms enable rapid target detection and are divided into two categories: one-stage detectors such as the YOLO series and SSD series, which offer fast detection speed but relatively lower accuracy; and two-stage detectors represented by the R-CNN series, which provide higher detection accuracy but poor real-time performance. YOLOv5 has gained favor among researchers due to its fast detection speed and good accuracy. Specifically, YOLOv5 incorporates Mosaic data augmentation, adaptive anchor calculation, and adaptive image scaling at the input stage, uses CSPDarknet53 based on CSPNet as its feature extraction network to reduce memory consumption within a certain range, and employs FPN and PANet structures in the processing output section to accelerate information flow between layers.

However, the traditional YOLOv5 network has a large number of parameters and suffers from no attention preference during feature extraction, applying identical weighting to features of different importance levels. This study focuses on natural environment apples, analyzing the effects of different coloring times post-bag-removal, different lighting conditions, and background presence/absence of white cloth on results. By performing lightweight improvements on the YOLOv5 detection algorithm and integrating a yield fitting network, an apple tree yield estimation model was established that takes image data as input to estimate apple tree yield, providing references for rational harvest personnel arrangement

during fruit harvest periods and technical support for intelligent agricultural equipment in modern orchard environments.

2.1 Image Data Acquisition

In-situ images were collected at the orchard base of Shandong Tongda Modern Agriculture Group Co., Ltd. in Yao Village, Guandao Town, Qixia City, Yantai, Shandong Province (37°16 N, 120°64 E). The apple variety was "Yanfu No. 3," with fruits harvested 16-22 days after bag removal. The acquisition platform was a self-built quadrotor UAV (Figure 1[Figure 1: see original paper]) equipped with a Raspberry Pi 4B as the core for image acquisition and storage, featuring a Cortex-A72@1.5GHz CPU, Broadcom VideoCore VI GPU, 8 GB RAM, and 128 GB storage capacity. The camera (Raspberry Pi Camera V2) had 5 megapixels, 30 Hz acquisition frequency, F2.35 maximum aperture, 3.15 mm focal length, and 65° field of view.

To reduce interference from other fruit trees, a white cloth measuring 4 m long and 3 m high was used as background, moving with the UAV (as shown in Figure 1(c)). During image acquisition, the UAV flew at a height of 1.5 m, 1.2 m from the trees. Images were captured under sunny conditions from three angles: front lighting, side lighting, and backlighting, both with white background and natural conditions, for training the yield estimation model and verifying its effectiveness in natural environments.

Images collected at different time points were used to analyze the effects of different coloring times and lighting conditions on the detection algorithm. The dataset included both whole-tree images and partial images. During operation, the detection algorithm only iteratively trained on annotated fruit regions. Since whole-tree and partial images captured by the UAV had similar fruit region sizes and distributions, using both as training data could improve algorithm generalization without losing detection accuracy.

Apples colored for 1 day appeared greenish, similar to leaf color; at 8 days, they began coloring, showing light red; at 15 days, they were fully colored. Image acquisition was conducted from October 3 to 17, 2020, every 7 days, between 10:00 AM and 4:00 PM, yielding three datasets at coloring stages of 1 day, 8 days, and 15 days. Tree yield data were collected when fruits had colored for 16 days (Figure 2[Figure 2: see original paper]).

2.2.1 Data Cleaning

To reduce the impact of duplicate images and fruitless images on model training, manual screening was used to delete duplicate images caused by UAV hovering and images without apples during attitude adjustment. After cleaning, partial data from the three coloring stages are shown in Figure 3[Figure 3: see original paper]. Different lighting condition data collected at different coloring times were used for comparative analysis with white cloth background data to verify the practical application effect of the proposed detection algorithm. Partial

apple images captured at different time points without white cloth background are shown in Figure 4[Figure 4: see original paper].

2.2.2 Dataset Division and Annotation

After data cleaning, 1,000 images were retained for each coloring stage (1 day, 8 days, and 15 days). From the white cloth background data, 300 images were randomly selected as Test Set 1, divided into three subsets (front lighting, side lighting, and backlighting) with 100 images each. From the non-white cloth background data, 300 images were selected as Test Set 2 without lighting condition subsets. The remaining images served as the training set.

Manual annotation was performed using labelImg software to select target fruits with bounding boxes containing position coordinates and category information. The annotation interface is shown in Figure 5[Figure 5: see original paper] (page 104). After annotation, the sample dataset was converted to standard PASCAL VOC2012 format. Image and bounding box quantities are shown in Table 1.

2.3 Yield Data Collection

Yield data were collected on October 18, 2020, from 9:00 to 17:00 when apples had colored for 16 days. Individual trees were numbered, and UAV images were captured before harvest. After harvest, fruits from each tree were placed in the same basket and weighed using an electronic scale. Subtracting the basket weight yielded individual tree yield. A total of 93 data groups were obtained, each containing tree images and corresponding yield. Sixty groups were used for yield fitting network training, 13 groups for network testing, 10 groups for yield model validation with white cloth background, and 10 groups for validation without white cloth background.

Scatter plots of different yield datasets are shown in Figure 6[Figure 6: see original paper]. Fruit quantity and tree yield showed certain linear correlation, with average fruit weight between 250-280 g, indicating good tree growth and no significant individual differences.

3 Apple Tree Yield Estimation Model

3.1 Overall Model Structure

The proposed fruit tree yield estimation model consists of two parts: a fruit detection algorithm and a yield fitting network (Figure 7[Figure 7: see original paper]). The fruit detection algorithm uses improved YOLOv5 for target detection on input tree images, outputting fruit quantity and total bounding box area. The yield fitting network takes these algorithm outputs as inputs and uses a BP neural network to fit tree yield. The detection algorithm was trained using the image training set, while the yield fitting network was trained separately using the yield training set. After training, the model can directly output corresponding tree yield from input tree images.

3.2.1 Lightweight-Improved YOLOv5

The YOLOv5 model originates from YOLO, which demonstrates good detection speed by regressing bounding box coordinates and categories at the output layer. The core YOLO concept divides input images into 7$×$7 grids, with the grid containing the target center responsible for prediction. Each grid predicts 2 bounding boxes that regress position coordinates and confidence values. A confidence threshold filters out low-confidence boxes, and non-maximum suppression (NMS) processes the retained boxes to obtain final predictions (Figure 8[Figure 8: see original paper]).

However, traditional YOLOv5 has numerous parameters and suffers from no attention preference during feature extraction, applying identical weighting to features of different importance. This study proposes replacing standard convolutions in the YOLOv5 feature extraction network with lightweight depthwise separable convolutions and introduces a Pooling Block Attention Module (PBAM) based on depthwise separable convolutions and visual attention mechanisms. PBAM is added to the YOLOv5 network to solve the no attention preference problem.

PBAM uses a compress-then-expand approach to enhance learning of key points sampled from shallow features. The module introduces a residual unit-like structure to prevent gradient vanishing or explosion in deep networks. PBAM maintains the same output feature map resolution as input, allowing it to be embedded into any network structure without modification, offering simple structure and convenient usage. By establishing inter-channel dependencies, it achieves adaptive calibration of channel-wise features. Figure 9[Figure 9: see original paper] shows the improved YOLOv5 algorithm block diagram, with red boxes indicating the improved components.

The fused YOLOv5 algorithm can reduce computational pressure from attention mechanism modules using depthwise separable convolutions while extracting deep feature maps with more important information from shallow features computed by convolutions, further extracting key information and improving overall detection performance.

3.2.2 Loss Function Calculation

The total loss function consists of three components: confidence loss (L_{conf}), classification loss (L_{cla}), and coordinate loss (L_{GIoU}). The improved YOLOv5 algorithm only deepens the network without affecting these functions, requiring no new loss function construction:

L_{total} = L_{conf} + L_{cla} + L_{GIoU}

Confidence and classification losses are calculated using cross-entropy methods (Equations (1)-(3)):

L_{conf} = λ{obj} {i=0}^{S^2} {j=0}^{B} [I_{ij}^{obj} ln(C_i) + (1-

Ĉ_i)ln(1-C_i)] + λ{nobj} _{i=0}^{S²} *{j=0}^{B} I*{ij}{nobj} ln(C_i)

L_{cla} = _{i=0}^{S²} *{j=0}^{B} I*{ij}{obj} [p̂_i(c)ln(p_i(c)) + (1-p̂_i(c))ln(1-p_i(c))]

Where S^2 is the number of grids, B is the number of bounding boxes predicted per grid, I_{ij}^{obj} indicates whether the j-th bounding box in the i-th grid has a target to predict, I_{ij}^{nobj} indicates whether it has no target, *λ{obj} and* λ{nobj} are weight coefficients for grids with/without targets, C_i and Ĉ_i are predicted and actual confidence values, c is the target category in the bounding box, p_i(c) is the predicted probability of category c when a target is detected in the i-th grid, and p̂_i(c) is the actual probability.

This study uses L_{GIoU} as the bounding box coordinate loss function (Equations (4)-(6)):

IoU = |A ∩ B| / |A ∪ B|

GIoU = IoU - |C − (A ∪ B)| / |C|

L_{GIoU} = *{j=0}^{B}* {i=0}^{S²} (1 - GIoU)

Where A is the ground truth box area, B is the predicted box area, and C is the smallest enclosing convex area of A and B.

3.3 Yield Fitting Network

Since the functional mapping between estimated fruit quantity, bounding box area, and yield is unclear and exhibits nonlinear characteristics, this study employs a BP neural network for yield fitting to enhance the model' s capability for complex pattern classification and multidimensional function mapping. Because the input includes bounding box area, image capture distance and camera parameters are crucial. This study maintained constant platform-to-tree distance and fixed camera parameters during acquisition to ensure yield measurement accuracy.

The network structure uses 3 fully connected layers, 1 ReLU activation layer, and 1 Sigmoid activation layer (Figure 10[Figure 10: see original paper]). The BP network has 2 input neurons corresponding to fruit quantity and total bounding box area output by the improved YOLOv5 algorithm, and 1 output neuron corresponding to tree yield in the image. Adding ReLU or Sigmoid activation functions in hidden layers increases network nonlinearity, accelerates training, solves gradient vanishing during backpropagation, and effectively reduces overfitting.

The number of hidden layer neurons lacks clear theoretical determination. This study first determined an initial value using empirical formulas (Equation (7)), then selected optimal values based on error performance during training. The final hidden layer neuron count was 15 or 11.

N_h = √(m + n) + a

Where N_h is the number of hidden layer nodes, m is the number of input neurons, n is the number of output neurons, and a is a constant between 1-10.

The main steps for data fitting using the BP network are: 1. **Data normalization**: To ensure stability, fruit quantity, bounding box area, and tree yield were normalized by dividing by normalization coefficients to map inputs and outputs to 0-1 range. 2. **BP network training**: Four steps: (a) initialize network weights, (b) forward propagation, (c) backward error propagation, (d) update network weights and neuron biases. 3. **Data denormalization**: To obtain actual yield values, predictions were denormalized by multiplying by corresponding coefficients to remap data to the original range.

3.4 Model Training

Model training consisted of two stages: first, training the fruit detection algorithm to predict fruit quantity and bounding box area; second, yield prediction based on BP neural network fitting of fruit quantity, bounding box area, and yield.

The model was built using PyTorch deep learning framework. Hardware configuration included AMD Ryzen7 4800H CPU@2.9GHz, 6GB NVIDIA GeForce GTX 1660Ti GPU, 16GB RAM, and 512GB SSD. The operating system was Windows 10 64-bit. The code compiler was PyCharm 2019.3.3 Community Edition, configured with CUDA 10.2 and cuDNN 7.6.5 for GPU acceleration.

4 Performance Analysis

4.1 Fruit Detection Algorithm Performance Analysis

This study uses mean average precision (mAP) as the overall evaluation metric. Precision (P) is the proportion of actual positive samples among predicted positives, while recall (R) is the proportion of predicted positives among actual positives. The precision-recall curve (P-R curve) can be plotted from their relationship. Average Precision (AP) is the area under the P-R curve, and mAP is the mean AP across all categories (Equations (8)-(11)):

P = TP / (TP + FP)

R = TP / (TP + FN)

AP = $_{0}^{1}$ p(r) dr

mAP = (1/N) _{i=1}^N AP_i

Where TP is true positives, FN is false negatives, FP is false positives, TN is true negatives, and p(r) is precision at different recall rates.

To verify the feasibility of the proposed improved YOLOv5 fruit detection algorithm, performance tests were conducted on detection speed and mAP for YOLOv5 variants with: (1) only depthwise separable convolution replacement,

(2) only attention mechanism module embedding, and (3) fusion of both improvements, compared against unimproved YOLOv5.

Detection Speed Test: Table 2 shows detection speeds for multiple images (averaged) and relative improvement rates based on the original algorithm speed. YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x represent four network structures with increasing depth and width. Replacing standard convolutions with depthwise separable convolutions significantly improved detection speed by 11.26%-17.28%. Embedding attention mechanism modules alone had minimal impact on speed, remaining basically equivalent to the original. Fusing both methods increased computational burden slightly compared to depthwise separable convolution alone, but still achieved maximum speed improvement of 15.37% (fastest speed: 31.44 ms). These results demonstrate that depthwise separable convolution provides substantial speed gains, while the fused approach maintains fast detection.

mAP Test: Table 3 compares mAP across multiple images. Original YOLOv5 mAP values were 89.83%, 90.75%, 92.07%, and 93.44% for s, m, l, and x variants, respectively. All improved versions except YOLOv5s with only depthwise separable convolution exceeded 90% accuracy. Depthwise separable convolution alone caused minor accuracy loss (within 1%). The maximum accuracy difference between highest and lowest was 4.35%, occurring with attention mechanism-only embedding, which improved accuracy by up to 3.56% (average 3.17%) compared to original. Attention modules help transform shallow features into deep features with more important information, learning inter-channel correlations to enhance important features and suppress secondary ones, effectively improving overall detection. The fused YOLOv5l and YOLOv5x achieved over 95% accuracy, with YOLOv5x reaching 96.27%.

White Cloth Background Test: Under natural lighting, different shooting angles affect mAP. Tests were conducted on improved YOLOv5 using apple images from different time points and lighting angles (Figure 11[Figure 11: see original paper]). Front-lit photos were bright and clear without obvious shadow changes; side-lit photos had clear layers with distinct object contours; backlit photos were dark and blurry with underexposure issues. The improved YOLOv5 achieved best results across all lighting conditions and time points, with highest mAP of 96.79% on the 15-day side-lit dataset. Average mAP across different test sets was 93.30%, outperforming YOLOv5, YOLOv3, and SSD. Side lighting produced best results, while backlighting performed worst because dark fruit and leaf colors made edges unclear, increasing detection difficulty. Algorithm performance improved with coloring time since 1-day fruits were greenish and easily confused with leaves, while 15-day fruits were brightly colored and easily distinguishable.

No White Cloth Background Test: To verify natural environment applicability, tests were conducted on 1-day, 8-day, and 15-day images without white background (Figure 12[Figure 12: see original paper]). The algorithm completed detection tasks with high recognition rates for nearby fruits. Nearby apples oc-

cupy more pixels with richer features and higher prediction confidence, while distant apples have fewer pixels and lower confidence. Table 5 shows mAP comparison without white cloth background. Compared to white cloth background tests, accuracy decreased somewhat at all coloring stages. However, for yield estimation, only nearby tree yields need measurement. The algorithm' s ability to filter background tree fruits reduces interference, making ultra-high detection accuracy unnecessary. The proposed fruit detection algorithm meets the requirement of recognizing nearby apples while filtering distant ones, making it suitable for natural background conditions.

4.2 Yield Fitting Network Performance Analysis

Table 6 presents parameters evaluating yield fitting network performance. Correlation coefficient (R) and coefficient of determination (R^2) measure correlation between predicted and actual yields (higher values indicate better correlation). Root mean square error (RMSE) measures prediction error (lower values indicate higher precision). Mean absolute error (MAE) and mean absolute percentage error (MAPE) reflect prediction deviation (lower values indicate better fit). R values were 0.8979 and 0.8864, and R^2 values were 0.7967 and 0.7982 for training and test sets, respectively, indicating high linear correlation and good curve fitting. RMSE values were 1.5317 kg and 1.4021 kg, MAE values were 1.1259 kg and 1.0253 kg, and MAPE values were 6.3372% and 6.2524% for training and test sets, respectively.

Figure 13[Figure 13: see original paper] compares predicted and actual yields on the test set. The model predicted tree yields well with stable accuracy across different yield samples, demonstrating good robustness. Test results show the model is applicable for pre-harvest yield measurement in natural environments.

4.3 Yield Estimation Model Testing

White Cloth Background Model Testing: Using the proposed apple tree yield estimation model with white cloth background validation images as input, predicted yields were output. Table 7 shows relative errors ranging from -6.13% to 3.05%. Only the 7th group had relatively larger error, with overall relative errors within 7%. The model effectively learned important features from image and yield data through coordinated operation of detection algorithm and yield fitting network, demonstrating good prediction performance for white cloth background images.

No White Cloth Background Model Testing: To verify natural environment applicability, validation images without white cloth background were used as input. Table 8 shows relative errors ranging from -12.71% to 8.28%. Compared to white cloth background results, relative errors were larger due to interference from other tree fruits in the background. The yield fitting network predicted yields by treating all detected fruits as current tree fruits, causing some deviation. However, only some feature-rich apples in background trees

were recognized, as most distant apples with fewer pixels could not be identified, making the impact limited. Relative errors remained within 13% overall, indicating good precision, effectiveness, and robustness across different backgrounds for natural environment apple tree yield estimation. Adding more samples could further improve recognition accuracy and yield estimation precision.

Conclusions

This study proposes an apple tree yield estimation model fusing a yield fitting network with an improved YOLOv5 fruit detection algorithm. Through dataset preprocessing, model training, and application, the following conclusions are drawn:

1. The improved YOLOv5 apple detection network, enhanced by replacing standard convolutions with depthwise separable convolutions and adding attention mechanism modules, solves the problems of no attention preference and parameter redundancy in feature extraction. Using image datasets as input, it estimates fruit quantity and total bounding box area. Test results show the algorithm improves both lightweight characteristics and accuracy, achieving up to 15.37% detection speed improvement and 96.79% maximum mAP. Tests under different datasets demonstrate that lighting conditions, coloring time, and background presence of white cloth affect algorithm accuracy.

2. Using estimated fruit quantity and bounding box area as inputs and actual yield as output, the yield fitting network was trained. Test results show training and test set R^2 values of 0.7967 and 0.7982, RMSE values of 1.5317 kg and 1.4021 kg, respectively, with small yield estimation errors.

3. The final yield estimation model was obtained by fusing the fruit detection algorithm and yield fitting network. Experimental results show the fruit tree yield model achieved relative error ranges within 7% with white cloth background and within 13% without white cloth background, proving the established apple orchard in-situ yield model has good precision and robustness. With more sample data, target recognition and yield estimation accuracy could be further improved.

References

[1] Jiamusi Agricultural School of Heilongjiang province, Suzhou Agricultural School of Jiangsu province. General introduction to fruit cultivation[M]. Beijing: China Agriculture Press, 2009.

[2] WANG S, ZHANG Y, GAO H, et al. Apple bagging cultivation technology[M]. Jinan: Shandong Science and Technology Press, 2006.

[3] PAPAGEORGIOU E I, AGGELOPOULOU K D, GEMTOS T A, et al. Yield prediction in apples using fuzzy cognitive map learning approach[J]. Computers

and Electronics in Agriculture, 2013, 91: 19-29.

[4] AGGELOPOULOU A D, BOCHTIS D, FOUNTAS S, et al. Yield prediction in apple orchards based on image processing[J]. Precision Agriculture, 2011, 12(3): 448-456.

[5] ZHOU R, DAMEROW L, SUN Y, et al. Using colour features of cv.' Gala' apple fruits in an orchard in image processing to predict yield[J]. Precision Agriculture, 2012, 13(5): 568-580.

[6] CHENG H, DAMEROW L, BLANKE M, et al. ANN model for apple yield estimation based on feature of tree image[J]. Transactions of the CSAM, 2015, 46(1): 14-19.

[7] CRTOMIR R, CVELBAR U, TOJNKO S, et al. Application of neural networks and image visualization for early predicted of apple yield[J]. Erwerbs-Obstbau, 2012, 54(2): 69-76.

[8] ROY P, KISLAY A, PLONSKI P, et al. Vision-based preharvest yield mapping for apple orchards[J]. Computers and Electronics in Agriculture, 2019, 164: 104896.

[9] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2016: 779-788.

[10] REDMON J, FARHADI A. YOLO9000: Better, faster, stronger[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2017: 7263-7271.

[11] REDMON J, FARHADI A. YOLO v3: An incremental improvement[EB/OL]. 2018. arXiv: 1804.02767v1.

[12] BOCHKOVSKIY A, WANG C, LIAO H. YOLOv4: Optimal speed and accuracy of object detection[EB/OL]. 2020. arXiv: 2004.10934.

[13] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]//European Conference on Computer Vision. Cham, Switzerland: Springer, 2016: 21-37.

[14] ZHANG S, WEN L, BIAN X, et al. Single-shot refinement neural network for object detection[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2018: 4203-4212.

[15] WANG D, ZHANG B, CAO Y, et al. SFSSD: Shallow feature fusion single shot multibox detector[C]//International Conference in Communications, Signal Processing, and Systems. Cham, Switzerland: Springer, 2019: 2590-2598.

[16] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2014: 580-587.

[17] GIRSHICK R. Fast R-CNN[C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway, New York, USA: IEEE, 2015: 1440-1448.

[18] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6): 1137-1149.

[19] ZHOU W, ZHU S. Research on the application of smart examination room solutions based on deep learning technology[J]. Information Technology and Informatization, 2020(12): 224-227.

[20] WANG F. Artificial intelligence detection and recognition algorithm for masks and helmets based on improved YOLOv5[J]. Construction and Budget, 2020(11): 67-69.

[21] WANG C Y, LIAO H Y M, YEH I H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[C]//The IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Seattle, WA, USA: CVPRW, 2020: 390-391.

[22] LIN T Y, DOLLAR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2017: 2117-2125.

[23] LIU S, QI L, QIN H, et al. Path aggregation network for instance segmentation[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2018: 8759-8768.

[24] LI D, YAN Z, SUN J. Study on classification and detection technology of river floating garbage based on UAV vision[J/OL]. Metal Mine: 1-11. [2021-06-20]. http://kns.cnki.net/kcms/detail/34.1055.TD.20210608.1117.005.html.

[25] SIFRE L, MALLAT S. Rigid-motion scattering for texture classification[J]. Computer Science, 2014, 3559: 1-12.

Note: Figure translations are in progress. See original paper for figures.