A Research on Perceived Farm Interaction System Based on Machine Learning

Huzhaorui Kang
College of Computer Science and Technology, Changchun Normal University, Changchun, China
kanghuzhaorui@126.com

ABSTRACT

This paper presents a solution to the challenges in agriculture by designing a machine learning based farm interaction system that can accurately detect the maturity of farm produce in real-time and remotely control the farm environment in a narrowband IoT (Internet of Things) environment. The study focuses on apple ripeness as an example, and utilizes a web crawler tool to gather images of young, growing, and ripe apples. These images were then processed using data enhancement methods to create a comprehensive data set. The YOLO-v3DenseNet target detection model was used to train and test the data set. Finally, a web system was designed using the Vue and Flask frameworks and deployed to a cloud server. The experimental test results indicate that the system allows users to upload photos of farm crops or access real-time videos or photos of farms collected through the cloud, while displaying their maturity and confidence levels. The YOLO-v3DenseNet model used in the system can effectively detect the maturity of crops, even when crops overlap or are partially obscured. The average detection accuracy of this model for target feature images of crop maturity exceeds 90%, and can be applied to real-world farm environments.

CCS CONCEPTS

• **Artificial intelligence**; • **Computer methodologies**;

KEYWORDS

YOLO V3, Narrowband IoT, Target Detection, Machine Learning, Smart Agriculture

ACM Reference Format:
Huzhaorui Kang. 2023. A Research on Perceived Farm Interaction System Based on Machine Learning. In *2023 7th International Conference on Big Data and Internet of Things (BDIOT 2023), August 11–13, 2023, Beijing, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3617695.3617711

1 INTRODUCTION

In recent years, agriculture has encountered numerous challenges with regards to resources, environment, and costs. The cultivation of fruits and vegetables has been lagging behind due to insufficient high-tech technology to support modern operations. This has resulted in two main effects: on the one hand, The agricultural supply chain faces challenges due to excessive intermediaries in the sale of agricultural products and inadequate information flow. On the other hand, The harvesting of agricultural products is heavily dependent on farmers. However, due to the imbalance between urban and rural development [1], urban youths seldom have the opportunity to visit the countryside and lack understanding of agriculture. To tackle these issues, it is crucial to integrate a fruit ripeness detection model with narrowband IoT technology.

In the field of fruit ripeness detection, traditional methods involve manually extracting features such as fruit color, texture, and shape, and then use image processing or shallow machine learning methods to build segmentation and detection model. However, these algorithms, including Otsu segmentation and SVM-based detection methods, have been found to have low detection accuracy, slow speed, and poor robustness and generalization when applied in complex environments. In recent years, experts and scholars have done research in the field of crop maturity detection. These approaches has shown stronger generalization capabilities compared to manually extracted features, and its powerful characterization capability has been demonstrated in studies such as that of Sigit Widiyanto et al. [2]. They used Faster R-CNN models to recognize tomato maturity, taking advantage of the model's support for image classification and object detection. The accuracy for classification in validation stage about 98.70% in average. For the object detection the model has confidentiality about 96.20% to detect the tomato maturity. Hongxing Peng et al. [3] aimed to develop a method for quickly and non-invasively determining citrus maturity. They achieved this by utilizing machine vision and Android mobile platform technology. Zhao et al. [4] achieved a 96% accuracy rate in detecting ripe tomatoes in a real-world environment by utilizing a combination of an AdaBoost classifier and a color classifier. Tian et al. [5] proposed the YOLO-v3 dense model, which outperformed both the original YOLO-v3 model and the Faster R-CNN model with VGG16 nets - the most advanced model for fruit detection. To address the limitations of conventional crop ripeness recognition techniques and the challenges in the agricultural supply chain, this paper proposes an interactive system that combines narrowband IoT technology with the YOLO-v3-DenseNet target detection algorithm for fruit ripeness recognition on farms. This paper's goal is to provide technical assistance for the establishment of smart farms, accelerate the landing of new agricultural businesses, and generate multiple benefits.

BDIOT 2023, August 11–13, 2023, Beijing, China

ACM ISBN 979-8-4007-0801-5/23/08…$15.00
https://doi.org/10.1145/3617695.3617711

2 SYSTEM COMPONENTS AND DESIGN PROCESS

2.1 System components

The system is comprised of six modules: a hardware module, a data set collection module, a training module, a prediction module, a cloud server module, and a front-end display module. The hardware module utilizes multiple sensors and other hardware devices to gather information on light, soil moisture, temperature and humidity, CO2 concentration, and crop conditions on the farm. The data collection module employs a web crawler tool to crawl images, which are then selected, enhanced, and annotated. The training module utilizes an improved YOLO-v3 model and generates a weights file. The prediction model utilizes the weights file to process both live photos and farm photos collected in the cloud. It then returns the coordinates of the bounding box vertices for the identified crops, along with the confidence level of their maturity classification. The cloud server module offers cloud control, storage, and monitoring, ensuring stability during the image recognition phase. The front-end display module utilizes Vue and Flask frameworks to create a separated web system. Finally achieve the interface display of ripeness detection results for live photos collected in the cloud.

2.2 Design process

To create an interactive system on the farm, the first step is designing the hardware. This hardware primarily utilizes NB-iot modules in the communication layer. The main goal of this layer is to transmit the information collected by the various sensors in the sensing layer over long distances. Therefore, providing on-demand, ubiquitous communication support for the system. Secondly, in order to achieve crop maturity recognition, images are first crawled through a crawler tool and irrelevant images are manually removed. Once the data set has been enhanced, labeled files are efficiently generated using the LabelImg image annotation tool. Next, the dataset was trained using a pre-existing model for target detection. By adjusting the parameters, a highly accurate apple maturity recognition model was obtained. The resulting h5 weight file, named best_model.h5, records the image features extracted during training and the learning information. The trained model is used to detect live video or images taken from the farm and display the results in the front-end interface.

3 TARGET DETECTION MODULE

3.1 Apple maturity algorithm based on improved YOLO-v3 model

3.1.1 Network Structure. The YOLO-v3 [6] detection model is used in this paper, with its network model diagram shown in Figure 1. The basic classification network used is Darknet-53, which has been found to have greater advantages over more advanced CNN network models like ConvNeXt and RegNet. However, due to the small batch characteristics of the apple maturity feature picture in this paper, the normalization of Darknet-53 is not applicable. Therefore, a new Densenet network [7] is introduced to enhance efficiency. To improve the recognition accuracy of the Densenet network, an attention mechanism is introduced before each dense block. This mechanism reduces the model's attention to distracting information, such as leaves, and allows it to focus on important features for improved recognition.

3.1.2 Prediction of Target Bounding Boxes in YOLO-v3. The prediction of the target bounding box in YOLO-v3 is similar to YOLO-v2, as it uses a Sigmoid function to estimate the central point's relative position in relation to the grid cell's upper left corner. The confidence in the bounding box is reflected in two ways. On the one hand is the probability that the bounding box contains the target. On the other hand is the accuracy of the bounding box, which is expressed by the ratio of the intersection and the union of the areas of the two rectangular boxes, denoted as IOU, the expression of which is shown in formula 1.

$$\mathrm{IOU}_{\mathrm{B_{GT}},\ \mathrm{B_P}} = \frac{\mathrm{B_{GT}} \cap \mathrm{B_P}}{\mathrm{B_{GT}} \cup \mathrm{B_P}} \tag{1}$$

Where, $\mathrm{B_{GT}}$ is the boundary box of ground truth. $\mathrm{B_P}$ is the predicted bounding box.

However, the loss function defined by IOU cannot reflect the size and quality of the intersection degree, and cannot predict the distance in the case of non intersection. Rezatofighi, H. et al. [8] proposed the definition of GIOU by introducing the minimum bounding rectangle of the prediction box and the real box to solve the above problem. Therefore, this paper chooses to use GIOU loss to regress the position of the target, which can be calculated using formula 2.

$$\mathrm{GIOU}_{\mathrm{B_{GT}},\mathrm{B_P}} = \frac{\mathrm{B_{GT}} \cap \mathrm{B_P}}{\mathrm{B_{GT}} \cup \mathrm{B_P}} - \frac{|\mathrm{B} \backslash (\mathrm{B_{GT}} \cap \mathrm{B_P})|}{|\mathrm{B}|} \tag{2}$$

Where, B is the smallest convex object surrounded by $\mathrm{B_{GT}}$ and $\mathrm{B_P}$.

3.2 Experimental environment configuration

This study utilizes the Jupyter editor and the Python programming language to conduct an experiment using the deep learning framework Tensorflow as the running environment for the algorithm. Tensorflow offers TensorBoard, a tool for visualizing the complex computational graph. The computational graphs are defined, constructed, and executed using the user-friendly and open source Python language. TensorBoard allows for the visualization of the loss value of the model and the accuracy of the algorithm after detection. Additionally, TensorBoard can display the statistical graph of the parameters, providing insight into the change pattern of the parameters.

The platform framework used in this paper, Tensorflow, is installed via Anaconda and run via a terminal. Anaconda is an open source software library that includes deep learning libraries like conda and Python. Additionally, Tensorflow is highly compatible with popular systems, including Linux, Mac, and Windows. Table 1 displays the configuration of the experimental environment.

3.3 Image data reprocessing

3.3.1 Image Data Acquisition. After completing the collection, irrelevant images were manually removed, qualified photos were processed and uniformly renamed to a prescribed format and coded according to the degree of fruit ripeness to automatically form a uniformly named batch of data. A total of 360 images were collected for this data set, representing images of young apples, growing apples and ripe apples, as shown in Figure 2.

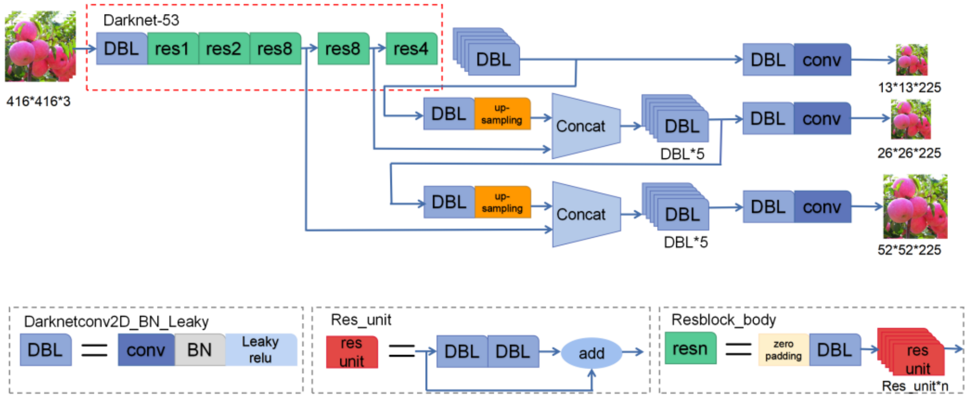

Figure 1: YOLO-v3 network model diagram

Table 1: Experimental environment

Experimental environment	Experimental configuration
Operating system	Windows 11
Memory	8.00 GB
Deep learning framework	Tensorflow
Programing language	Python3.6.5
Development tool	Jupyter

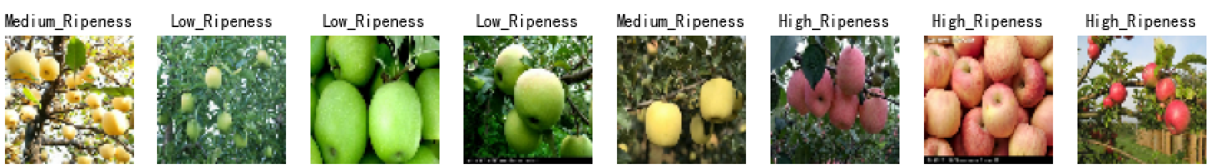

Figure 2: Example of an apple ripeness data set

Table 2: The number of images after using the data enhancement methods

	Original data	Brightness	Contrast	Crop	Flip	Rotation	Total
Number of young apple images	120	240	120	120	120	360	1080
Number of expanding apple images	120	240	120	120	120	360	1080
Number of ripe apple images	120	240	120	120	120	360	1080

3.3.2 Enhancement of Image Data. In order to make the apple data set more complete, this paper uses a series of data enhancement methods to expand the data set obtained using the crawler tool, such as brightness enhancement, contrast enhancement, mirroring, random rotation, cropping, etc. After processing, 3240 images of apples at three different stages of maturity were obtained by manually filtering valid images. The number of images after using the data enhancement methods is shown in Table 2.

Considering the inconsistent lighting intensity of the orchard, the model processes feature information differently when picking under different lighting intensities, and enhancing brightness and contrast can improve the model's generalization ability. Furthermore, the location of the apples in the field orchard can impact their ripeness detection. To account for this, this paper utilized mirroring and random rotation methods to pre-process our dataset. While improving detection accuracy, this data set can be expanded. In

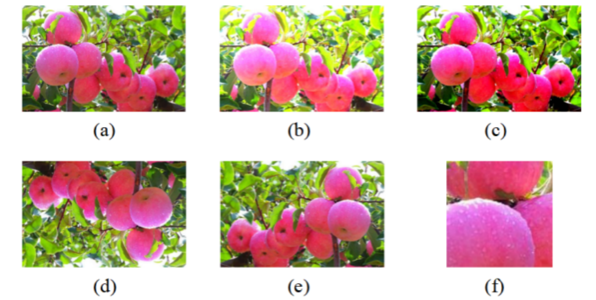

Figure 3: Sample data set pre-processing

Figure 4: Example of data set annotation

addition, the apple images collected in the growing environment are prone to multiple fruit overlap and leaf occlusion, which cannot be accurately identified to apples. By cropping the apple data set, it facilitates feature learning during network model training. The samples pre-processed for this data set are shown in Figure 3.

3.3.3 Image Annotation. In apple maturity detection, it is particularly important to label all the target objects in the data set. In this paper the image annotation tool LabelImg is used. Before the data set is annotated the required annotation file type needs to be selected. The YOLO-v3 network model requires an annotation file in txt file format. During the manual labelling process, the target area to be trained for recognition is selected by creating a rectangular box. An example of the annotation of the data set is shown in Figure 4.

3.4 mAP

MAP represents the average accuracy. If it is a category of objects, the detection accuracy is the mAP value. For the detection of three types of objects in this article, such as young apples, growing apples, and mature apples, taking the average detection accuracy of the three types of objects is the average accuracy of this article. The calculation formula is shown in equation 3.

$$\mathrm{mAP} = \frac{\sum_{\mathrm{C}=1}^{\mathrm{C}} \mathrm{AP}(\mathrm{C})}{\mathrm{C}} \tag{3}$$

Where C is the number of categories.

3.5 Loss function

The evaluation metric in this paper uses the change curve of the loss value of the loss function with the number of iterations to evaluate the improvement efficiency and optimisation effect of the model.

Table 3: Different algorithm detection performance

Methods	mAP%	Time/s
Faster-R-CNN	83.79	0.065
YOLO-v1	61.27	0.270
YOLO-v2	72.65	0.107
YOLO-v3	82.33	0.087
YOLO-v3-DenseNet	92.53	0.061

The loss function of YOLOv3 is divided into three parts, which are target confidence loss($\mathrm{Loss}_{\mathrm{conf}}$), target localization offset loss($\mathrm{Loss}_{\mathrm{giou}}$) and target classification loss($\mathrm{Loss}_{\mathrm{cla}}$), and the overall loss function is the weighted sum of the three.

$$\mathrm{Loss}_{\mathrm{conf}} = \begin{cases} -\alpha\left(1-\mathrm{y_P}\right)^{\gamma} \times \log \mathrm{y_P},\ \mathrm{y_{GT}} = 1 \\ -\left(1-\alpha\right)\mathrm{y_P}\gamma \times \log\left(1-\mathrm{y_P}\right), \mathrm{y_{GT}} = 0 \end{cases} \tag{4}$$

$$\mathrm{Loss}_{\mathrm{giou}} = 1 - \mathrm{GIOU}_{\mathrm{B_{GT}},\mathrm{B_P}} \tag{5}$$

$$\mathrm{Loss}_{\mathrm{cla}} = \mathrm{C_{GT}} \log \mathrm{C_P} - \left(1-\mathrm{C_{GT}}\right)\log\left(1-\mathrm{C_P}\right) \tag{6}$$

$$\mathrm{Loss}_{\mathrm{total}} = \alpha_1 \mathrm{Loss}_{\mathrm{giou}} + \alpha_2 \mathrm{Loss}_{\mathrm{conf}} + \alpha_3 \mathrm{Loss}_{\mathrm{cla}} \tag{7}$$

Where, $\mathrm{y_P}$ and C_P donate the predicted confidence and the predicted category, respectively. y_{GT} and C_{GT} individually represent the ground truth confidence and the ground truth category. α_1, α_2 and α_3 are balance coefficients.

4 SYSTEM ARCHITECTURE AND SYSTEM IMPLEMENTATION

4.1 System architecture

The system uses Vue and Flask frameworks to implement a front and back-end separated web system and deploy it to a cloud server. Among the physical architecture, the server operating system is Centos7. The specific development tools are divided into web development tools and model development tools, where the web development tools include Notepad++, Visual Studio Code, and the model development tools include Anacoda3, Pytorch, TensorFlow2.4.1, Pycharm 3, Jupyter notebook, TensorBoard, LabelImg. Based on the above development tools and basic architecture, the overall architecture of this system is designed as shown in Figure 5.

4.2 Experimental comparison and analysis

By conducting experiments on the Faster R-CNN algorithm in the R-CNN series and the YOLO-v1, YOLO-v2,YOLO-v3, and YOLO-v3-DenseNet algorithms in the YOLO series for apple maturity detection in this article, a comparison of the detection performance in Table 3 is obtained. It can be clearly seen from the table that YOLO-v3-DenseNet has significantly improved detection accuracy and approached real-time detection speed, while the YOLOv5 algorithm released in 2020 has significantly improved detection speed, It is not very helpful for this article to mainly improve the accuracy of identifying apple fruits, so this article continues to use the YOLO-v3 algorithm for optimization and improvement.

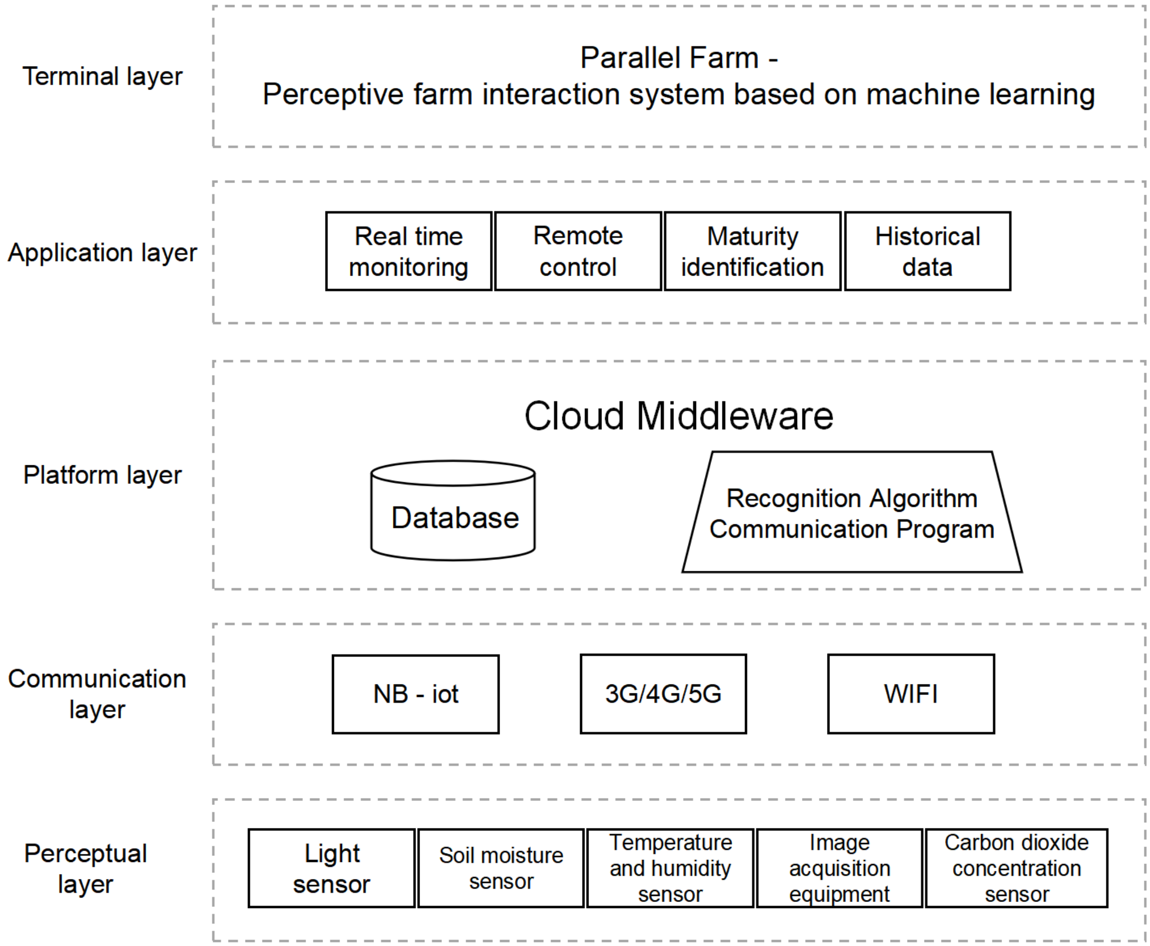

Figure 5: Overall structure

4.3 System implementation

After the process of system analysis, process design, training models and deployment of cloud servers, the final realisation of the machine learning based Perceptive Farm Interactive System achieves the following functions: (1) remote control; (2) real-time monitoring; (3) crop maturity recognition; and (4) automatic collection of crop information to provide data sources for subsequent big data analysis.

Users, as well as farm administrators, can detect the ripeness of fruits through the live images collected from the farm and transmit this information to the cloud platform layer. Users can also upload the images of fruits to be detected in this system by taking images of fruits in the field, and the upload interface is shown in Figure 6.

After the image data transmission is completed, the Recognition Effect Feedback Interface pops up automatically. The interface provides intelligent feedback based on the recognition of the acquired image:

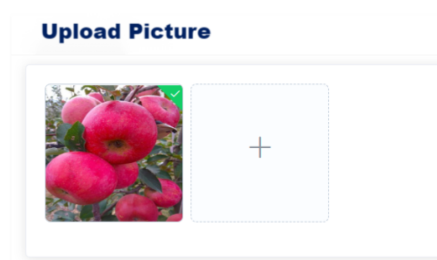

Figure 6: Upload interface

When the uploaded image does not contain the fruits of crops or the size is too small or obscured, or network interruption, transmission termination and other conditions, as well as poor uploaded image quality, too bright or too dark environment, the system will feed back to users the "unrecognized, it is recommended to re-upload" prompt.

Figure 7: Recognition effect feedback interface

When the target object exists in the uploaded image, and the network transmission is normal and the uploaded image quality is good, the corresponding recognition results are given according to the model algorithm to display the maturity of apple and its confidence, as shown in Figure 7.

5 CONCLUSION

- In response to the challenges currently faced by agriculture and the limitations of traditional fruit maturity recognition methods, a perceptual farm interaction system for crop fruit maturity recognition was designed by combining narrowband Internet of Things technology with target detection algorithm of machine learning technology YOLO-v3-DenseNet, which includes hardware module, data collection module, training module, prediction module, cloud server module and front-end display module. Rapid and accurate identification and detection of Apple maturity has been achieved.
- The system is stable and has good robustness. According to the results, it can recognize Apple maturity with an accuracy of over 90%.
- The research only collected data on a single variety, so there are some limitations in this system. In order to meet the actual needs in future research, it is necessary to increase representative fruit images of varieties, such as Jonakia and Snake fruit. Additionally, it would be beneficial to subdivide the maturity stages in order to increase recognition and detection accuracy and better meet the actual needs.

REFERENCES

[1] M. R. M. Kassim, "IoT Applications in Smart Agriculture: Issues and Challenges," 2020 IEEE Conference on Open Systems (ICOS), Kota Kinabalu, Malaysia, 2020, pp. 19-24, doi: 10.1109/ICOS50156.2020.9293672.
[2] S. Widiyanto, D. T. Wardani and S. Wisnu Pranata, "Image-Based Tomato Maturity Classification and Detection Using Faster R-CNN Method," 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2021, pp. 130-134, doi: 10.1109/ISMSIT52890.2021.9604534.
[3] H. Peng, Y. Shao, Y. Hu, D. Wu, Y. Chen and G. Lin, "Maturity Analysis of Citrus Based on Machine Vision and Android Mobile Platform Technology," 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xiamen, China, 2018, pp. 319-322, doi: 10.1109/ICITBS.2018.00088.
[4] Zhao, Y., Gong, L., Zhou, B., Huang, Y., & Liu, C. (2016). Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. Biosystems Engineering, 148, 127–137. doi:10.1016/j.biosystemseng.2016.05.
[5] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Computers and Electronics in Agriculture, 157, 417–426. doi:10.1016/j.compag.2019.01.012
[6] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
[7] Huang, G., Liu, Z., Maaten, L. van der, & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.243
[8] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2019.00075