DOI: 10.54254/2755-2721/53/20241330

A study on fruit classification using convolutional neural network for image recognition technology

Haipeng Shi

College of civil engineering and architecture, Zhejiang University, Zhejiang, China

3180102315@zju.edu.cn

Abstract. Artificial intelligence is being incorporated into more aspects of our daily lives. Agricultural production in China is at a critical juncture due to the rising fruit production and dwindling labor force, necessitating the adoption of mechanization and intelligent systems. Image processing is particularly suitable for this field. Image classification technologies in fruit categorization are examined in this research. The results of this study can help improve the development of intelligent, lightweight machinery for agricultural output. This, in turn, will enhance the efficiency of fruit cultivation, harvesting, and trading and alleviate labor constraints. Two popular convolutional neural network models, namely ResNet50 and MobileNetV2, were employed in this study. The study utilized two optimizers: SGD and Adam. The evaluation results revealed that the ResNet50 model, employing SGD optimization, achieved the highest accuracy of 95.57%. Despite its lower accuracy of 92.19%, the MobileNetV2 model demonstrates higher efficiency than ResNet50 due to its lower hardware requirements, rendering it suitable for operation on compact devices.

Keywords: Fruits classification, machine learning, deep learning, CNN.

1. Introduction

Numerous industries, including transportation, education, agriculture, and various services, use computer vision extensively [1-3]. Computer vision is a widely used technical tool in both the industrial and agricultural sectors. It uses automated fruit harvesting, sorting equipment, and supermarket fruit scanning [4]. Both the agricultural and industrial sectors routinely identify and categorize fruits. Fruit identification and classification improve product packaging efficiency on farms, whereas they speed up the process of fruit shelving in supermarkets. As a result, identifying and classifying fruits is essential to achieving these goals.

With its ranking as the world's third-largest country by land area and the highest global population, China exhibits remarkable scale. By the end of 2019, the cultivated area of various fruits in China had experienced rapid growth, covering 12.277 million hectares and yielding an annual output of 190.38 million tons [5]. This extensive production places significant demands on fruit identification and classification. However, production efficiency in China remains low, primarily due to the persistent reliance on traditional manual farming methods and inadequate use of advanced scientific technologies, leading to significant human errors. Furthermore, the aging labor force and a declining birth rate in China have spurred the inevitable trend of labor automation to replace human workers. The research and implementation of automatic fruit image recognition can relieve the labor-intensive nature of fruit classification, resulting in labor and financial resource savings, enhanced work efficiency, and more time and energy available for other tasks.

This paper aims to investigate the fruits within the Fruit 360 dataset and present a deep learning-based approach for recognizing fruit varieties and facilitating the identification of multiple fruit types. This paper will develop and enhance models utilizing ResNet50 and MobileNetV2, renowned deep-learning networks, to achieve precise classification and recognition of various fruit varieties. Furthermore, the effectiveness of the models will be assessed.

2. Literature review

Presently, numerous methods have been proposed by researchers for automatically identifying different types of fruits. Classical fruit classification and recognition methods typically entail extracting features using manually designed feature extraction techniques, encompassing aspects such as size, shape, color, texture, and other relevant characteristics from fruit images. These features are subsequently combined to construct one or more classifiers, enabling the attainment of automatic fruit classification and recognition, as depicted in Figure 1 [6]. Zhang et al., for example, extracted and merged SURF and color moments from pictures [7]. Using the K-means clustering technique and SVM technology, they got great results with a 94% identification rate. Nevertheless, this method does not yield satisfactory performance when handling online downloaded image datasets with diverse backgrounds and poses. Nevertheless, this method does not yield satisfactory performance when handling online downloaded image datasets with diverse backgrounds and poses. Thinh et al. examined three feature extraction techniques: edge detection, RGB histogram, and HOG [8]. The efficiency of three categorization models was also investigated: Random Forest, KNN, and SVM. The SVM model paired with HOG features provided the best accuracy (96%), according to their analysis. Complete Local Binary Pattern (CLBP) use as a textural trait for identifying fruits and vegetables was suggested by Tao et al. [9]. They combined color and texture information, using a nearest neighbor classifier to categorize fruits and vegetables, and took into account variations in lighting intensity. However, it was difficult to distinguish between fruit and vegetable representations with intricate backgrounds.

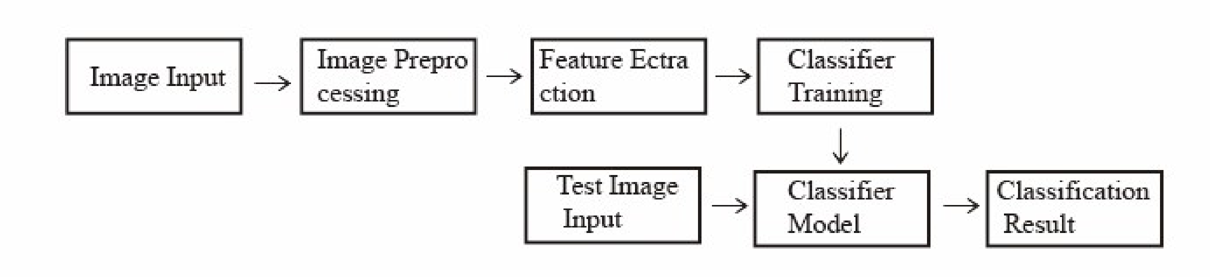

Figure 1. Traditional fruit image classification and recognition system flowchart [6].

Deep learning has revolutionized feature extraction by automatically extracting image features through network structures, eliminating the need for manual control. This advancement enables more comprehensive and precise feature extraction. As an example, Saranya et al. compared convolutional neural networks (CNNs) against KNN and SVM [10]. The CNN fared better than the conventional algorithms, with a significantly higher accuracy of 96.49%. A 6-layer CNN created by Lu et al. was created specifically for fruit classification [11]. The model had an accuracy of 91.44% after being trained on a dataset including 1800 photos of 9 different kinds of fruit. Rectified linear units (ReLU) were used in the construction of an 8-layer deep CNN Wang et al. used parameter calibration [12]. Each fully linked layer was implemented before a dropout layer. The use of data augmentation methods to avoid overfitting led to a remarkable overall accuracy of 95.67%. The model beat five current approaches, including conventional machine learning approaches and a sophisticated CNN methodology. In addition to fine-tuning the pre-trained deep learning model VGG-16, created exclusively for the visual geometry group (VGGNet), Hossain et al. incorporated a six-layer CNN model [13]. According to the results, the refined classical network model performed admirably in classifying fruits and vegetables.

From a classification methods standpoint, while certain classifiers based on traditional algorithms exhibit commendable accuracy and robustness, they frequently necessitate substantial engineering work for image feature extraction. Convolutional neural networks (CNNs) obviate the requirement for manual design of feature extraction procedures by automatically acquiring features from raw image data. This capability enables the generation of high-quality convolutional feature maps, while significantly reducing the need for manual involvement. Moreover, CNNs can mitigate the impact of noise using convolutional and pooling layers, thereby ensuring stability in recognizing image variations and facilitating accurate identification across diverse environmental conditions. Nevertheless, current models often encounter difficulties when classifying fine-grained varieties that belong to the same fruit category. Furthermore, these models impose high hardware demands during operation, while their training and storage requirements necessitate substantial memory capacity. Such factors substantially escalate the costs associated with research. Consequently, these models are typically incompatible with portable devices like mobile phones and are not extensively employed by small and medium-sized enterprises or individual consumers.

3. Dataset

The fruit dataset utilized in this study, generated by Mihai Oltean and made available to the public, is called Fruits 360. It is updated in real-time regularly. In May 2020, the dataset was last updated. The 90,483 photos in Fruits 360 each have a 100x100 pixel size. This dataset contains 67692 pictures for training and 22688 photographs for testing. It encompasses 131 distinct fruit types, and each image exclusively depicts a single fruit or vegetable, as depicted in Figure 2. Mihai Oltean captured the initial images of fruits by positioning them on a low-speed motor axis (3rpm) against a white paper background. A 20-second video was recorded using a Logitech C920 camera. However, the initial images are characterized by non-uniform backgrounds due to variations in lighting conditions. To mitigate this, the flood-fill algorithm was employed to process the images.

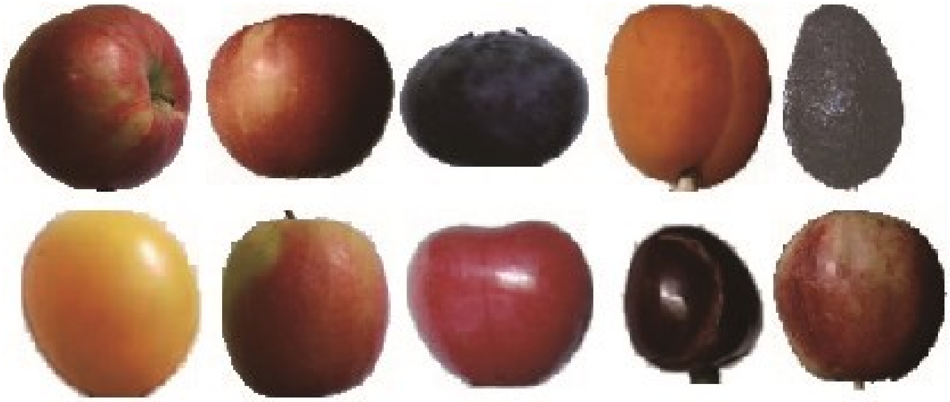

Figure 2. Partial fruit varieties in the dataset.

4. Deep learning approach

4.1. Convolutional neural network

4.1.1. Introduction to Convolutional Neural Networks. CNNs were initially multilayer neural networks created especially for jobs requiring image recognition. LeCun et al. used CNNs to successfully tackle the problem of reading handwritten digits in 1995 [14]. Figure 3 shows the LeNet5 network model they developed. LeNet5 comprises 8 layers and serves as the foundational architecture for contemporary convolutional networks [15]. Input layers, hidden layers, and output layers are all features shared with conventional neural networks in terms of their general structure. Convolutional layers and pooling layers, also known as subsampling layers, are two essential building elements that make up the hidden layers, the main part of CNNs. This structure also includes regularly used completely connected layers. Within the framework of convolutional neural networks, the convolutional layers and pooling layers serve as essential modules for feature extraction within the hidden layers. This network model gradually adapts the weight parameters within each layer via backpropagation, while minimizing the loss function through gradient descent. As a result, with repeated training, the model's accuracy is increased repeatedly. In the lowest levels of the convolutional neural network, convolutional layers and pooling layers alternate. However, the upper layers are analogous to the entirely connected and hidden layers seen in logistic regression classifiers and traditional multilayer perceptrons. A classifier is the last output layer.

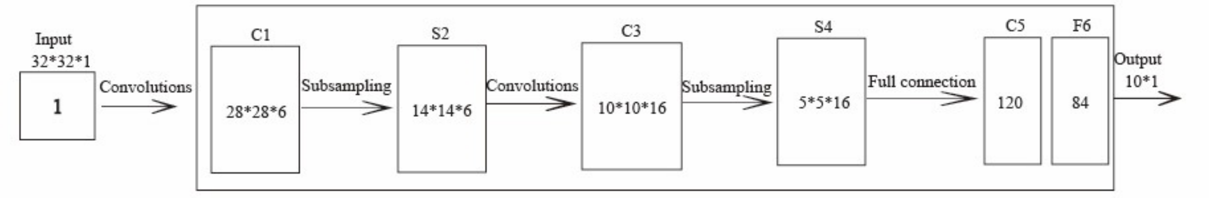

Figure 3. The structure of the LeNet5 network [15].

4.1.2. Introduction to ResNet50. CNNs have consistently outperformed in image classification challenges. Furthermore, increasing the network's depth enhances its effectiveness in feature extraction. However, as the network's depth increases, the problem of vanishing gradients becomes more apparent, making network optimization harder. In light of this, He et al. created the Residual Convolutional Neural Network (ResNet), a network design that enhances photo classification job performance while obtaining more depth [16]. The ResNet family of networks is now one of the most widely utilized network designs in the field of image-related applications. ResNet is made up of stacked residual blocks, as seen in Figure 4 [16]. In addition to weight layers, a residual block has skip connections, which link the input x straight to the output. H(x) represents the original mapping, whereas F(x) represents the residual mapping. Furthermore, the skip connections make it easier to transmit features between layers, reducing the barrier faced by disappearing gradients to some extent. By stacking residual blocks, ResNet may generate network levels with up to 152 layers. The Residual Network has demonstrated notable effectiveness in picture categorization tasks. Among the Residual Network architectures, ResNet50 and ResNet101 represent the most renowned network structures. These models demonstrated outstanding performance in the 2015 ILSVRC competition, ultimately claiming the championship.

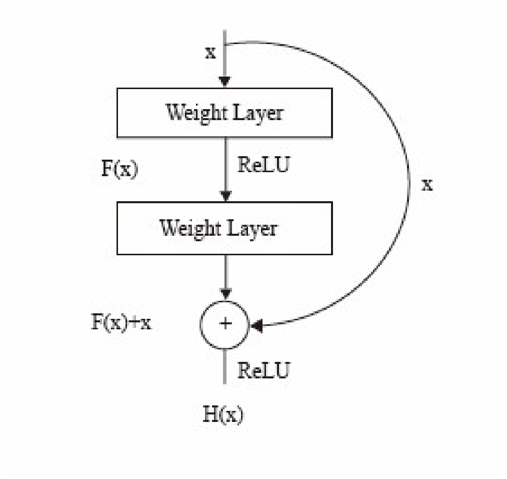

Figure 4. Residual block structure [16].

4.1.3. Introduction to MobileNetV2. With increasing network depth, deep network models exhibit improved performance. Nevertheless, this presents challenges including a substantial growth in model parameters and slower inference speed. In light of these challenges, Google introduced a lightweight network called MobileNet V1 in 2017 [17]. MobileNet V1 was primarily designed to utilize depth-wise separable convolutions in place of conventional convolutions. The conventional convolution is broken into two components in depth-wise separable convolutions: the depth-wise convolution and the point-wise convolution [17]. As seen in Figure 5, this split considerably decreases computational complexity, which frequently falls between 1/8 and 1/9. MobileNet achieves a harmonious balance between performance and efficiency, exhibiting traits such as low latency and low power consumption. It proves to be ideal for deployment on mobile devices such as smartphones, which possess comparatively limited computational capabilities and hardware configurations when compared to computers. Through substantial reduction of computational complexity, while striving to retain optimal performance, MobileNet has found extensive applications in the realm of image recognition.

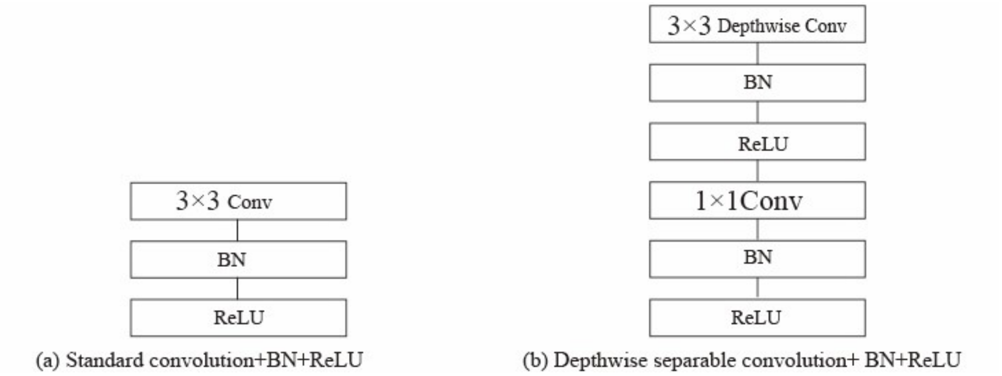
3×3 Conv
BN
ReLU
(a) Standard convolution+BN+ReLU
3×3 Depthwise Conv
BN
ReLU
1×1Conv
BN
ReLU
(b) Depthwise separable convolution+ BN+ReLU

Figure 5. Depthwise separable convolution [17].

MobileNet V2 enhances the initial MobileNet architecture through the incorporation of linear bottleneck layers and inverted residual structures, as illustrated in Figure 6 [18]. The linear bottleneck layer eliminates the final non-linear activation function found in the conventional bottleneck layer. In terms of channel dimensions, the inverted residual structure diverges from the traditional residual structure by first increasing and then decreasing the channel count. The increased number of channels enables the extraction of more features, thereby enabling the model to strike an optimal balance between parameter count, computation time, and performance.

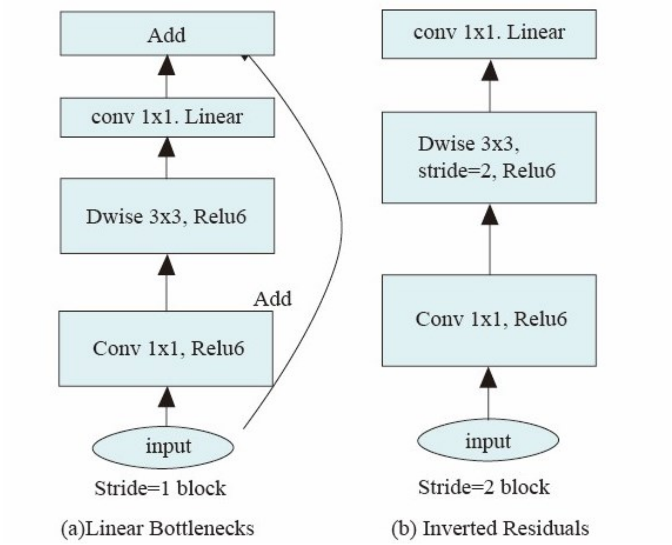

Figure 6. MobileNetV2 Convolutional Block [18].

4.2. PyTorch Deep Learning Framework

Within the realm of deep learning, numerous remarkable learning frameworks exist, and this article selects PyTorch, which was introduced by Facebook's Artificial Intelligence Research team (FAIR) in 2016. PyTorch strives to offer a rapid, adaptable, and dynamic deep learning framework. It shares a design philosophy reminiscent of Python, emphasizing enhanced readability and simplicity over unwarranted complexity. Notably, PyTorch has made several daring design decisions, with the most significant being the selection of a dynamic computation graph as its foundation. The dynamic computation graph fundamentally diverges from the static computation graph utilized in other frameworks, primarily due to its capacity to accommodate runtime modifications to the graph. Consequently, this attribute renders PyTorch highly flexible in managing intricate models, making it

4.3. Model optimization based on gradient descent method

Gradient descent serves as the prevailing optimization algorithm for objective functions within the field of deep learning. To locate the local minima of the function, it initially computes the gradient direction of the loss function by taking its derivatives. Subsequently, it iteratively traverses in the opposite direction of the current point of the loss function by a predetermined step size, progressively converging towards the local minimum. The fundamental principle of gradient descent entails that as the function value approaches the target value, the corresponding gradient diminishes, resulting in a deceleration of the descent. Stochastic gradient descent (SGD) and the Adam algorithm were selected as the optimal optimization strategies for this investigation. Stochastic gradient descent substitutes the global gradient with the gradient of a random sample, enabling swift network convergence and the acquisition of a favorable local optimum within a condensed timeframe. Nonetheless, there is a possibility of converging to a local extreme point, resulting in diminished accuracy. The Adam algorithm amalgamates concepts from the Momentum algorithm and the RMSprop algorithm. It determines the gradient's mean and variance before determining the update step size to dynamically change the learning rate. This algorithm exhibits a low computational cost, necessitates less memory, as well as being resilient to gradient scaling and diagonal re-scaling. Consequently, it proves to be well-suited for processing sparse data and non-stationary objectives. Presently, it stands as one of the most optimal algorithms for gradient descent performance.

4.4. Evaluation standards

Predictions and corresponding results are used for classification, producing four possible outcomes: True Negative (TN) is the ability to identify and classify negative events; False Positive (FP), often known as the false positive rate, denotes wrongly classifying instances of negativity as positive; False Negative (FN), also known as the false negative rate, denotes mistakenly classifying positive cases as negative, whereas True Positive (TP) denotes accurately classifying positive instances as positive. Precision, Recall, F1 Score, and Accuracy, are performance indicators used to evaluate classification models.

$$Recall = TP(TP + FN)^{-1} \tag{1}$$

$$Precision = TP(TP + FP)^{-1} \tag{2}$$

$$Accuracy = (TP + TN)(TP + TN + FP + FN)^{-1} \tag{3}$$

$$F1\,Score = 2 \times (Precision \times Recall)(Precision + Recall)^{-1} \tag{4}$$

5. Results and discussion of the experiment

After determining the various parameters of the model, the classifier is executed, and the outcomes are presented in Table 1. Given the dataset's extensive array of fruit categories, weighted averages are employed to represent each performance metric accordingly.

Table 1. Model classification results.

		Training set		*Test set*				
Model	optimizer	Accuracy (%)	Loss	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Loss
ResNez50	**SGD**	99.99	0.0063	95.57	96.37	95.43	95.30	0.1543
	Adam	99.28	0.0456	89.82	92.06	89.82	89.25	0.3282
MobileNetV2	**SGD**	99.26	0.0088	92.19	94.53	91.89	91.46	0.2784
	Adam	97.95	0.1042	87.54	90.42	87.51	86.61	0.4043

By evaluating the accuracy of the model's training set, it can be observed that irrespective of the model type or optimizer employed, high accuracy is consistently achieved. Nevertheless, the accuracy of MobileNetV2 with Adam optimizer stands at 97.95%, marginally lower than the counterparts exceeding 99%. All four models exhibit minimal loss values. The employment of the SGD optimizer yields superior accuracy and reduced loss values when juxtaposed with the Adam optimizer. These findings indicate that ResNet50 outperforms MobileNetV2 in terms of performance on the training set, and that the Adam optimizer performs better during network training.

Results from the test set demonstrate that both models and optimizers achieve an accuracy of more than 85%, demonstrating remarkable performance. Regarding accuracy, ResNet50 excels by approximately 2% compared to MobileNetV2, whereas SGD surpasses Adam by approximately 5%. Consequently, the combination of ResNet50 with SGD optimization yields the highest recognition accuracy. Precision-wise, both models and optimizers attain accuracies surpassing 90%, signifying commendable performance. ResNet50 surpasses MobileNetV2, while SGD optimization outperforms Adam. Analyzing recall and loss values, it is evident that the ResNet50 model exhibits higher recall rates and lower loss rates compared to MobileNetV2. These findings suggest that the ResNet50-based recognition model excels in performance, showcasing robust generalization capabilities with minimal overfitting. Furthermore, the SGD optimizer demonstrates a superior average recall rate and lower loss value compared to Adam optimization, indicating enhanced performance of the SGD-trained network. In conclusion, the ResNet50-based recognition model demonstrates superior performance across multiple indicators, while SGD optimization yields superior outcomes. This discrepancy may arise due to the inability of MobileNetV2, being a lightweight network, to rival the performance of ResNet50 when employed on extensive datasets. However, MobileNetV2 presents advantages in terms of shorter training duration, heightened efficiency, and reduced computational demands, rendering it adequate for specific applications. Convolutional neural networks also perform remarkably well at differentiating between subcategories of the same variety of fruit. They successfully categorize fruits with similar appearances but different types.

This study possesses certain limitations. While the Fruits 360 dataset employed exhibits notable diversity, the image quality predominantly consists of pristine white backgrounds for each fruit image. Nevertheless, numerous images within the dataset exhibit high similarity, captured from varying rotational angles. Consequently, this particular fruit dataset falls short of fulfilling authentic real-world business requirements. During practical applications, it is improbable for fruit recognition to involve background removal from images before recognition procedures.

6. Conclusion

The ResNet50 and MobileNetV2 models are used in this paper to demonstrate a fruit categorization system based on computer vision. The Fruit 360 dataset is used by the system to do experiments. The experiment findings show that on the validation dataset, the ResNet50 model outperforms the MobileNetV2 model. The fruit classification system using the ResNet50 model achieved an accuracy of 95.57%, while the system using the MobileNetV2 model achieved an accuracy of 92.19%.

Despite taking longer to create, the classifier model does away with the requirement for tedious feature extraction and selection found in conventional machine learning techniques. Despite its lower accuracy, the MobileNetV2 model is lightweight and well-suited for deployment in computer vision-based systems. Future research should focus on refining and improving this model to enhance its accuracy and enable automation in fruit orchards. Furthermore, it is essential to choose image datasets that better represent real-world environments for training purposes.

References

[1] Bhargava A and Bansal A 2018 Journal of - King Saud University - Computer and Information Sciences **33** 243-57

[2] Koresh J 2019 Journal of Innovative Image Processing **33** 243-57

[3] Illeperuma G and Sonnadara D 2017 4th Int. Conf. on Electrical Engineering, Computer, Science and Informatics (EECSI) 1-6

[4] Naranjo-Torres J, Mora M, Hernández-García R, Barrientos RJ, Fredes C and Valenzuela A 2020 Applied Sciences **10** 3443

[5] Xiaofeng N 2021 China Fruit News **38** 42

[6] Chen B and Chen G 2022 Computer Era **7** 62-5

[7] Zhang Z and Ju Z 2020 Electronic Science and Technology **33** 41-5

[8] Thinh N V, Nhi A N T Y, Huy T G, Hoang Khoa N and Cam N T 2021 IEEE Int. Conf. on Machine Learning and Applied Network Technologies (ICMLANT) 1-6

[9] Tao H, Zhao L, Xi J, Yu L and Wang T 2014 Transactions of the CSAE **30** 305-11

[10] Saranya N, Srinivasan K, Pravin Kumar S K, Rukkumani V and Ramya R 2020 Computational Vision and Bio-Inspired Computing (Advances in Intelligent Systems and Computing vol 1108) 79-89

[11] Lu S, Lu Z, Aok S and Graham L 2018 IEEE 23rd Int. Conf. on Digital Signal Processing (DSP) (Shanghai, China) 1-5

[12] Wang SH and Chen Y 2020 Multimed Tools Appl **79** 15117-33

[13] Hossain M S, Al-Hammadi M and Muhammad G 2019 IEEE Transactions on Industrial Informatics **15** 1027-334

[14] Lecun Y and Bengio Y 1995 The Handbook of Brain Theory and Neural Networks **15** 1027-334

[15] Lai X 2021 Master's Thesis of Zhejiang A&F University 07

[16] He K, Zhang X, Ren S and Sun J 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 770-778

[17] Howard A G, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M and Adam H 2017 arXiv preprint arXiv 1704.04861

[18] Sandler M, Howard A, Zhu M, Zhmoginov A and Chen L C 2018 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition 4510-4520