Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230652

Performance comparison of different convolutional neural networks for vegetable and fruit recognition

Yuxuan Chen[1,†], Shengkai Pan[2,†], and Haoyu Wang[3,4,†]

[1] Sun Yat-sen University, Guangzhou, 510275, China
[2] Wuhan University of Technology, Wuhan, 430070, China
[3] The University of New South Wales, Sydney, 2052, Australia

[4] z5242863@unsw.edu.au
[†] These authors contributed equally.

Abstract. Image classification, a significant research problem in the computer vision community, aims to assign different types of images to a certain category in a fixed category set according to different information features reflected in images. With the continuous improvement of living standards, people's daily demand for accurate identification of food categories (such as vegetable, fruit, etc.) is growing. Early vegetable and fruit recognition mostly relied on manual features and machine learning algorithms, with low recognition accuracy and weak generalization ability. Thanks to the rapid evolution of the convolutional neural network, vegetable and fruit recognition based on depth learning has made breakthroughs in accuracy and speed. In this paper, three representative convolutional neural networks are introduced around vegetable and fruit recognition, and their performance differences are quantitatively compared on different data sets to explore the boundaries of their applications. In addition, we summarized the research problems in vegetable and fruit recognition and discussed its future development direction.

Keywords: image classification, deep learning, computer vision, vegetable and fruit.

1. Introduction

Nowadays, the pursuit of a healthy lifestyle is becoming a national trend, and diet is an indispensable part of it. Presently, the recording of daily diet mainly depends on manual input. With few applications, low accuracy, and a small recognition range of photo recognition records, the mainstream recording software lacks the recognition and classification function of vegetable and fruit. Vegetable and fruit recognition aims to predict the categories contained in the image (such as apples, tomatoes, etc.) according to different information features reflected in the image. In practical applications, the difficulties of vegetable and fruit recognition mainly lie in large within cluster variation, scale changes, viewpoint changes, object occlusion, light changes and background clutter. Therefore, the development of a set of high precision vegetable and fruit image classification technology has important academic value and application value.

The early traditional image classification method mainly includes feature extraction, feature processing and classifier design. By scanning the image with a certain stride, the computer can extract its features. Common feature extraction algorithms include Histogram of Oriented Gradient, Harris, while common features include intuitive features (such as brightness, texture, and colour), gray

DOI: 10.54254/2755-2721/5/20230652

statistical features (such as perimeter, area, etc.). Meanwhile, feature processing can eliminate features with less information as far as possible. Principal Component Analysis and Singular Value Decomposition are the most commonly used methods. Finally, we design the classifier according to the learned features. At present, the commonly used classifiers are Support Vector Machine, Random Forest, etc. However, since traditional methods mainly rely on manual extractor design, their generalization ability and robustness are poor, which cannot meet the needs of practical applications.

In the last few years, vegetable and fruit recognition based on the Convolutional Neural Network (CNN) has made continuous breakthroughs in accuracy and speed, largely due to the rapid evolution of deep learning technology[1]. Deep Learning means learning from experience by obtaining the internal logic of samples in the learning process and applying it to new data. Its final goal is to enable the machine to independently learn knowledge, analyze data and summarize laws like the human brain. Compared with traditional methods, deep learning has a more complex network structure and the ability to automatic feature extraction. It is more applicable to complex and large amounts of data and brings higher classification accuracy, stronger generalization ability and higher robustness. Deep learning technology has been extensively used in the field of image classification and has made remarkable achievements, of which CNN is a crucial part. CNN began to be studied in the 1980s, mainly includes the input layer, convolutional layer, pooling layer and fully connected layer. It solves the problems of the fully connected neural network that slow calculation speed and getting overfitted easily caused by too many parameters. Up to now, the study on CNN is still continuing.

On this basis, we apply three representative models based on CNN to the classification of vegetable and fruit images, e.g. AlexNet, ResNet and EfficientNet, and compare the accuracy of the three models and their pre-trained models. Through quantitative experimental analysis, we discussed the advantages and disadvantages of different models and tried to analyze the reasons for the differences in the accuracy of the three models.

2. Methods

2.1. Convolutional neural network

A typical convolutional neural network architecture consists of convolutional layers, pooling layers, fully connected layers, activation functions, as well as Softmax classifier.

The convolutional layer's role is to extract the image's features. By sliding the convolution kernel (also known as weight filter) on the input image or the input feature map, the result of matrix multiplication and addition in the sliding process is the output feature map. Convolution has two advantages: one is local perception. Second, weight sharing. These two advantages give convolution a strong feature extraction ability and migration ability. At the same time, the number of parameters of convolution operation is significantly reduced compared with the full connection. The expression of the convolution kernel is as follows (1):

$$Z_{i,j}=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} K_{m,n}V_{i+m,j+n}+b \tag{1}$$

where $Z_{i,j}$ stands for the output feature map's value at position (i,j), $K_{m,n}$ for the convolution kernel's data at position (m,n), $V_{i+m,j+n}$ for the image's data at location (i+m, j+n), and b for the bias term.

The pooling layer is also known as the down-sampling layer. The feature size extracted after the convolutional layer is still significant, and direct training with it can easily produce an overfitting phenomenon. So one idea is to aggregate statistics for features at different locations and select representative values to represent the original features, which introduces the pooling layer. Additionally, the pooling approach has two benefits: first, it reduces the computational cost of the network by altering the size of the input feature map; second, it extracts the key characteristics by compressing the features. In this research, the maximum pooling layer's approach is used to minimize the feature vectors generated by the convolutional layer. The expression of the maximum pooling function is as follows (2):

$$y_{k,i,j}=\max_{(m,n)\in R_{i,j}} x_{k,m,n} \tag{2}$$

DOI: 10.54254/2755-2721/5/20230652

where $y_{k,i,j}$ represents the maximum pooled output value in the k^{th} rectangular region $R_{i,j}$, and $x_{k,m,n}$ represents the element at (m,n) in the k^{th} rectangular region $R_{i,j}$.

Convolution and pooling are linear computations, and so are their combinations. Therefore, nonlinear elements must be introduced to make the algorithm have good representation ability. Therefore, the activation function is introduced into the neural network. The activation functions of AlexNet and ResNet networks used in this paper are both ReLU functions, while EfficientNet uses Swish activation functions. The parameters of the network can be updated quickly in the process of network training, which is one of the merits of the ReLU function. And this method can make the forward propagation speed very fast so that the whole calculation efficiency is ideal. It is widely used in convolutional neural networks. The formula of the ReLU activation function is (3):

$$f(x)=max(x,0) \tag{3}$$

The Swish activation function is non-monotonic and smooth in comparison to the ReLU activation function. Ramachandran demonstrated that Swish could achieve higher test accuracy than ReLU on intense and mobile models. The following is the Swish function formula (4):

$$f(x)=x\cdot sigmoid(\beta x) \tag{4}$$

In convolutional neural networks, full connections emerge in the last few layers to integrate and extract useful information. Convolutional layers, pooling layers, and activation functions are some examples of structures that translate the source data to the feature vector space, while the role of the fully connected layer is to integrate the learned features and then map them to the sample label space. The formula of the fully connected layer is as follows (5):

$$y=Wx+b \tag{5}$$

where W stands for weight and b for bias.

The Softmax classifier maps the eigenvectors of the input neural networks into the (0,1) space, while the sum of these values is 1, and the output values can be understood as probability values. Therefore, when output results, choose the highest probability value as the classification result. The following is the Softmax function (6):

$$Softmax\ (z_i)=\frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \tag{6}$$

where z_i is the output value of the i^{th} node, and the total number of categories is K.

2.2. Pre-training

Training the deep neural network is always time-consuming and computationally expensive, especially on challenging tasks or using large-scale datasets. As a compensation method, leveraging a pre-trained model on related tasks is a kind of widely used transfer learning technique[2] in the deep learning domain. The parameter in the pre-trained model can be considered as prior knowledge learned from the previous task, which can provide a strong generalization ability to related tasks. Instead of learning from scratch on the new task, exploiting prior knowledge can accelerate the training process and benefit the optimization of the related task.

After training on the previous task, the convolutional layers in the model contain different levels of prior knowledge. Freezing part of convolutional layers, especially the layers close to the input layer, can retain the learned underlying information. By fine-tuning the remaining unfrozen layers using the new dataset, the model is capable of extracting the specific feature in the new task while fully utilizing the prior knowledge. In addition to this, depending on the property of the new task, freezing all convolutional layers is also optional. In this case, for classification tasks, only the final fully-connected layer will be trained to adapt to the new task. Compared with training from scratch, using the parameter in the pre-trained model as initialization is more optimal than random initialization. As a result, the training process will be faster and easier to converge than without using the transfer learning technique.

In this paper, in addition to training three kinds of deep neural network models from scratch, we also adopt their corresponding pre-trained versions, respectively. In doing so, the performance of different deep neural network models can be evaluated in two dimensions. In addition to the horizontal

DOI: 10.54254/2755-2721/5/20230652

comparison between different models and between their pre-trained versions, the performance gap between each model and its pre-trained version is also available to analyze the generalization ability of the pre-trained model on the vegetable and fruit image classification task.

2.3. AlexNet

AlexNet[3] is the founding work of the modern deep convolutional neural network. This model won the ImageNet Image Recognition Challenge in 2012. The validity of CNN in complex models is proved. The network architecture of the AlexNet model is illustrated in Figure 1. For the first time, ReLU, Dropout, and LRN algorithms have been effectively implemented in CNN using AlexNet's 8-layer convolutional neural network.

The three highlights of the AlexNet network are the change of the Sigmoid activation function to the more straightforward ReLU activation function, which does not require exponentiation like the Sigmoid activation function. In order to prevent model overfitting and hence manage the model complexity of fully linked hidden layers, dropout is utilized to arbitrarily deactivate some neurons. By adopting the values fed by local neurons to lessen the correlation between surrounding neurons, the LRN layer was created as a competitive mechanism by relying on the idea of lateral inhibition in neurobiology. This was done in an effort to increase the model's ability to generalize.

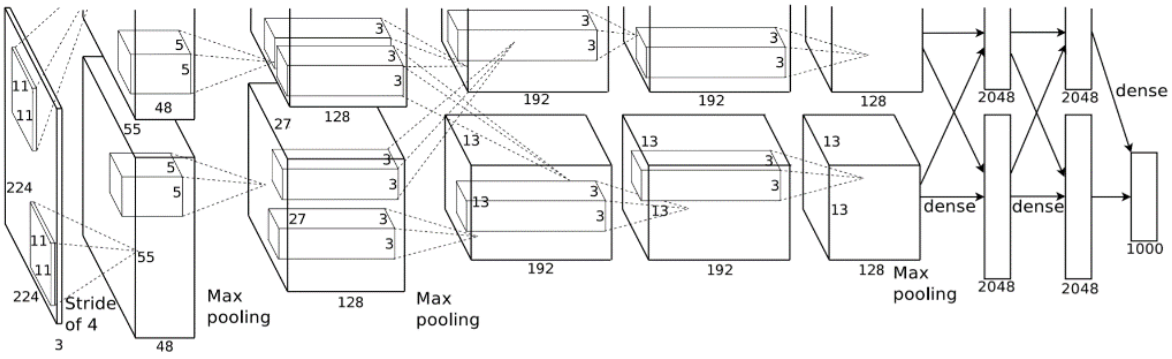

Figure 1. Structure of AlexNet network[3].

2.4. ResNet

The gradient explosion, gradient vanishing, as well as degradation problems always occur during the training process of deep neural network. These problems become more serious over the increase of the network depth. Several techniques, such as normalized initialization and batch normalization, has been leveraged to mitigate the gradient explosion and gradient vanishing problem. However, adding more intermediate layers cannot solve the degradation problem, and even causes higher training loss. The training accuracy will still decline rapidly after rising steadily to saturation. To alleviate the degradation problem, based on the architecture of VGGNet[4], ResNet was proposed in [5]. The authors verified that the deep residual learning framework using the shortcut connection is beneficial to optimization. The training accuracy can be improved without introducing more parameters and computation.

As shown in Figure 2, let x and $H(x)$ denote the input and output of a building block, respectively. In ResNet, we let the building block fit the residual mapping $F(x) = H(x) - x$, rather than the output $H(x)$. By inserting the identity shortcut connection x, the residual building block is formulated, which is prone to optimize compared with the original building block. In some cases, fitting the residual to 0 is easier than fitting it to x when an identity mapping approximates to the optimal function. Such shortcut connection will not introduce additional parameters. Furthermore, when the input and output of a building block do not share the same dimension, another kind of shortcut connection with a linear project $x_{out} = W_{x_{in}}$ can be utilized to ensure a consistent input and output dimensions. The Stochastic Gradient Descent (SGD) can be used as usual to trained the residual building block. As a result, the

DOI: 10.54254/2755-2721/5/20230652

computational complexity and training efficiency will not be negatively affected if the computational cost of the addition operation $F(x) + x$ is ignorable.

Considering the available computational resources and the large-scale VegFru dataset, in this paper, we train an 18-layer ResNet from scratch to solve the vegetable and fruit image classification task. For comparison, another 18-layer pre-trained ResNet using the ImageNet dataset is exploited to verify the effectiveness of the image prior. There are 8 residual building blocks in the 18-layer ResNet. Each residual building block composes 2 convolutional layers with filter size 3×3 and a shortcut connection. The number of filters is adjusted according to the output dimension. Furthermore, the batch normalization follows closely behind each convolutional layer and ReLU is used as the activation function. Moreover, an average pooling layer and a fully-connected layer are employed. The full-connected layer contains 73 neurons to adapt the vegetable and fruit image classification task.

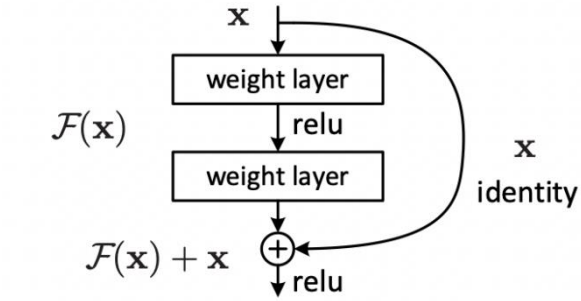

Figure 2. Residual Building Block[5].

2.5. EfficientNet

EfficientNet[6] is a network model proposed by Google in May 2019. It has achieved an amazing 84.4% accuracy on ImageNet, higher than all networks in previous years. In addition, EfficientNet also reduces the order of magnitude of parameters and FLOPS, which greatly increases the network efficiency. The construction of EfficientNet model mainly includes two steps: first, generate the baseline model EfficientNet-B0; Secondly, unlike traditional methods, which only scale depth, width and resolution separately, EfficientNet uses a model composite scaling method on the basis of EfficientNet-B0, that is, it synchronously scales the three dimensions to obtain EfficientNets.

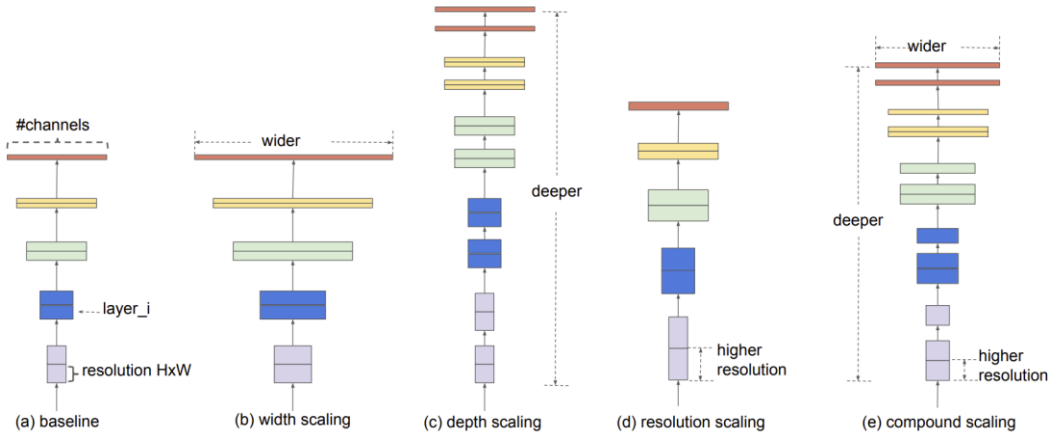

Figure 3. The network structure of EfficientNet[6].

EfficientNet-B0 is composed of 9 stages, of which Stage 1 is a common convolution layer with filter size 3×3; Stage 9 contains a convolution layer with filter size 1×1, a pooling layer and an FC classification layer; Stage 2 to Stage 8 are the same Mobile inverted bottleneck revolution (MBConv) structure, which is also an important innovation part of EfficientNet. The network structure of MBConv

DOI: 10.54254/2755-2721/5/20230652

is divided into five layers, one for each one 1×1 dimension-ascending convolution layer, one $k \times k$ Depthwise layer, one SE module, one 1×1 dimension -reducing convolution layer, a Dropout layer. Wherein, Depthwise Conv layer will divide the 3×3 convolution into two convolutions, thus reducing the number of parameters. The SE module is a kind of attention mechanism module[7], which is composed of a pooling layer and two FC layers. It can enhance the weight of features with good effects and weaken the weight of features with poor effects according to the loss of learning feature weights. In the module, firstly, perform the squeeze operation to compress the two-dimensional feature channels into real numbers; Then perform the enumeration operation to generate the corresponding weight of each feature channel; Finally, the Reweight operation is performed to multiply and weight the weights with the previous features, so as to realize feature re calibration.

3. Experiments and results analysis

3.1. Dataset

VegFru[8] is a domain-specific dataset for image classification tasks, which composes 160,731 vegetable and fruit images. It consists of two subsets, Veg200 containing vegetable images and Fruit92 containing fruit images. Veg200 dataset contains 15 sup-classes and 200 sub-classes with 91,117 vegetable images. Fru92 dataset contains 10 sup-classes and 92 sub-classes with 69,614 fruit images. For each sub-class, the number of images ranges from 200 to 2,000.

VegFru is a large-scale dataset, while the available computational resources are limited. Therefore, in this paper, we use 73 sub-classes with the largest number of images as the dataset used in the experiment. The selected dataset composes 72,086 images and each sub-class contains at least 700 images.

Figure 4. Some vegetable and fruit images from the VegFru dataset[8].

3.2. Evaluation metrics

In most classification tasks, Precision, Recall, and F-measure are mainly used as evaluation metrics to quantifying the model. Precision is, among all instances predicted as a specific class, the proportion of instances that truly belong to this class, while Recall represents the proportion of instances predicted as a specific class among all instances that truly belong to this class. F-measure is calculated as the weighted harmonic average of these two indicators, which considers these two indicators comprehensively. For evaluating the performance of each model in multi-dimensions, in this paper, we calculate Precision, Recall, and F-measure in micro, macro, and weighted views, respectively. Micro average treats each instance equally and computes the global average across all classes, while macro average treats each class equally and calculates the unweighted mean of all classes. By contrast, weighted average assigns weights to each class depending on their data. It reflects the weighted mean of all classes.

DOI: 10.54254/2755-2721/5/20230652

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{7}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{8}$$

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

Cross Entropy Loss is another effective indicator for classification task, which comes from the information theory. It measures the performance of a classification model via computing the cross entropy between the output and target. For multi-class classification tasks, the output of each class is a probability value between 0 and 1. The cross entropy loss will decrease along with the predicted probability of the target class increases. During the test process, the cross entropy loss will be calculated to reflect the performance of the model.

$$\text{CorssEntropyLoss} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{M}(w_m \cdot y_{n,m} \cdot \log \hat{y}_{n,m}) \tag{10}$$

where N and M represent the number of images and classes, respectively. w is the class weight. y and $\hat{y}$ denote the output and target, respectively.

3.3. Implementation details

PyTorch, Sklearn, Opencv, Numpy, as well as Matplotlib toolboxes are used for implementation. The experiment runs on Google Colaboratory with NVIDIA Tesla P100-PCIE-16GB for acceleration. We use 6 different models including AlexNet, 18-layers ResNet, EfficientNet, and their corresponding pre-trained versions. The structure of each model is described in detail in Section 2. The pre-trained versions are all based on the Imagenet dataset.

Each class in the dataset is randomly split into two subsets with a 9:1 training-test split ratio. The training dataset and test datasets consist of 64,912 and 7,174 vegetable and fruit images, respectively. The test dataset is used for finally evaluating the performance of each model. We use cross validation method in training process to reduce the interference of randomness to the experimental results. The validation dataset is designed for tuning the hyper-parameters and analyzing the errors of each model.

For data pre-processing, all images are resize into a fixed resolution 224×224. Furthermore, we utilize random flip as well as random rotation methods for data augmentation. Each channel of the image is normalized with designated mean and standard deviation. Moreover, to facilitate the training process, we use the Adaptive Moment Estimation (Adam) and Step Learning Rate (StepLR) as the optimizer and scheduler to introduce the momentum, weight regularization, as well as learning rate decay.

3.4. Quantitative evaluation

The evaluation indexes in subsection 2 of Section 3 are used to conduct statistics on the results of training from scratch and pre-training of the three models. Table 1-6 lists the evaluation indices obtained using AlexNet, ResNet, and EfficientNet, respectively.

Table 1. Evaluation metrics of AlexNet training from scratch.

AlexNet	Micro Average	Macro Average	Weighted Average
Precision		66.59%	67.14%
Recall	66.46%	65.61%	66.46%
F-measure		65.64%	66.36%
Loss		1.12	

Table 2. Evaluation metrics of pre-trained AlexNet.

AlexNet	Micro Average	Macro Average	Weighted Average
Precision		73.28%	73.44%
Recall	72.22%	72.46%	72.61%
F-measure		72.54%	72.84%
Loss		0.98	

DOI: 10.54254/2755-2721/5/20230652

Table 3 Evaluation metrics of ResNet18 training from scratch.

ResNet18	Micro Average	Macro Average	Weighted Average
Precision		76.12%	76.96%
Recall	76.24%	74.44%	76.42%
F-measure		75.29%	76.71%
Loss		1.06	

Table 4 Evaluation metrics of pre-trained ResNet18.

ResNet18	Micro Average	Macro Average	Weighted Average
Precision		83.66%	84.35%
Recall	84.29%	81.97%	84.11%
F-measure		82.91%	84.33%
Loss		0.69	

Table 5 Evaluation metrics of EfficientNet training from scratch.

EfficientNet	Micro Average	Macro Average	Weighted Average
Precision		91.52%	91.77%
Recall	91.78%	91.38%	91.78%
F-measure		91.38%	91.71%
Loss		0.27	

Table 6 Evaluation metrics of pre-trained EfficientNet.

EfficientNet	Micro Average	Macro Average	Weighted Average
Precision		93.70%	93.71%
Recall	93.62%	93.29%	93.62%
F-measure		93.36%	93.53%
Loss		0.20	

Overall, EfficientNet outperforms the other two models in all metrics from the six tables obtained. In detail, precision, recall, and F-measure values are approximately equal in each model, and F-measure is more focused on the lower value of precision and recall. This indicates that each of the models has about the same ability to correctly predict positive sample precision and predict positive sample fullness for the vegetable and fruit classification task, with EfficientNet having the best of these two abilities.

Based on the analysis of the differences among the three models, we draw the following conclusions:

(1) For each model, the pre-trained model performs better than the scratch: its accuracy is improved by 2%~8%, and the fitting speed is faster. This is because pre-training is essentially a kind of transfer learning. Using the pre-trained model is to make a series of fine-tuning on the model that has been trained by others for our own goals and tasks. The three models used in this experiment are all pre-trained on ImageNet, which has lots of advantages. First of all, ImageNet contains a sufficient number of pictures in a complete range of categories. ImageNet contains more than 20000 categories and 14 million pictures, all of which have been manually labeled. In this way, the general features of the data can be learned more easily, and the pre-trained model has a strong generalization ability and is not easy to be overfitting. Secondly, for AlexNet, ResNet, EfficientNet and other networks, the number of parameters is large and the network structure is complex, so it is difficult to achieve a good training effect using ordinary data sets. Therefore, using the pre-trained model on ImageNet is a good choice.

(2) From Table 1 to Table 6, we can see that EfficientNet has the best training effect and the most ideal evaluation indicators, followed by ResNet and AlexNet. This may be related to the different network structures of the three models. As mentioned earlier, AlexNet is composed of 5 convolution layers and 3 full connectivity layers. It deepens and widens the model through the stack of convolution layers. When it was proposed, its biggest highlight was the adoption of ReLU activation function, which accelerated the gradient attenuation in the training phase. However, ResNet model not only surpasses

DOI: 10.54254/2755-2721/5/20230652

AlexNet in the number of layers, but also introduces residual building block on the basis of ReLU activation function, which solves the problem of accuracy decline with the deepening of the network. This also makes ResNet better than AlexNet.

Different from ResNet and AlexNet, EfficientNet not only considers the network depth, but also considers the simultaneous scaling of the three dimensions of depth, width and resolution. Before that, these three parameters of the model were obtained through manual parameter tuning, but this is probably not the optimal solution. In this regard, EfficientNet searches for a set of optimal network parameters based on NAS (Neural Architecture Search). This may be the reason why EfficientNet is better than ResNet and ResNet is better than AlexNet.

(3) For training epoch, AlexNet is greater than EfficientNet and EfficientNet is greater than ResNet. But the initial learning rate adopted by AlexNet is very low, which is only 0.0002, while the initial learning rate adopted by the other two networks is 0.1. This may make AlexNet and ResNet are still underfitting after the training, resulting in a low accuracy.

4. Conclusion

This paper compares the differences between AlexNet, ResNet, and EfficientNet to find the model most suitable for solving the classification problem of vegetable and fruit. The main work is as follows:

To find a more suitable approach to solve the fruit classification problem from three typical classification models for people's need for daily dietary records for target classification. Seventy-three vegetable and fruit images from VegFru The selected dataset was trained using one of the commonly used deep learning frameworks, PyTorch, and then a test dataset was used to test the network to achieve the classification effect. Three networks, AlexNet, ResNet, and EfficientNet, use the same dataset to train from scratch and corresponding pre-training, respectively. According to the experimental findings, the EfficientNet model performs better at classifying fruits and vegetables.

The paper selected one of the three models as the most suitable method to solve the vegetable and fruit classification problem and analyzed the reasons why these models have different results for the same task, but there are still shortcomings, and improvements can be made in the following aspects.

(1) Increase the number of models to compare. In this study, only the differences among the three models for solving the vegetable and fruit classification problem were compared, and then other typical vegetable and fruit classification networks, such as GoogLeNet[9], DenseNet[10], and U-Net[11], could be compared and analyzed with EfficientNet on this basis.

(2) Increase the number of sub-classes used for training. In this study, only the sub-classes including no less than 700 images are used. Later, we can try to gradually increase the number of training sub-classes, including sub-classes with fewer images, to improve the model's capacity for generalization.

References

[1] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), pp.436-444.

[2] Pan, S.J. and Yang, Q., 2009. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), pp.1345-1359.

[3] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), pp.84-90.

[4] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[5] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[6] Tan, M. and Le, Q., 2019, May. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

DOI: 10.54254/2755-2721/5/20230652

[8] Hou, S., Feng, Y. and Wang, Z., 2017. Vegfru: A domain-specific dataset for fine-grained visual categorization. In Proceedings of the IEEE International Conference on Computer Vision (pp. 541-549).

[9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition(pp. 1-9).

[10] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[11] Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.