Journal of Research in Multidisciplinary Methods and Applications
http://www.satursonpublishing.com/
ISSN: 2957-3920

Volume 2, Issue 3, March 2023

Capsicum Recognition Based on Python and Convolutional Neural Networks

Wang Qing'e[1], Zhu Zhenglong[1*], Ye Zhenhuan[1], Tang Maoyin[2], Jiang Wenchang[1], Qian Tingting[1]

1. College of Engineering, Zunyi Normal University, Zunyi 56300, China;

2. Department of Automobile, Guizhou Aerospace Vocational and Technical College, Zunyi 56300, China

Correspondence author: Zhu Zhenglong

Abstract: This paper first briefly introduces the Python language and TensorFlow architecture, normalizes, calibrates, and divides the collected images, and then analyzes the processing process of the AlexNet model, adopts the VGG model and causes the data of the transfer learning method to be identified, to get 90% accuracy, using cross-entropy training loss function, the loss value is 0.3.

Keywords: Pepper Identification; Convolutional Neural Networks; Transfer Learning

1 Introduction

At present, pepper has become one of the main vegetable crops in China. However, due to the low degree of mechanized operation of pepper harvesting [1], the manual picking cycle is long, the cost is high and the labor intensity is large, which is not conducive to the development of the pepper planting industry chain. In recent years, the research of pepper harvester has made certain achievements [2]-4], but it is still lagging behind Western developed countries, in order to further solve the problem of pepper picking, this paper identifies and studies mature Chaotian pepper.

In recent years, artificial intelligence has made breakthroughs in the field of image recognition, and image recognition technology based on convolutional neural networks has become a research hotspot and widely used. The formation of convolutional neural network is inspired by the hierarchical working mode of the human brain optic nerve in biology, and has the characteristics of weight-value sharing, unsupervised learning to extract target features, and hierarchical structure [12]. The advantages of convolutional neural networks will greatly reduce the computational workload in pepper recognition. In order to improve efficiency, Python language will be used as the development language for network construction, and the convolutional neural network model will be built using the TensorFlow deep learning framework.

Python is an object-oriented interpreted computer programming language with a concise and easy-to-use language, high platform portability, and rich and powerful third-party packages. With the continuous development of Python versions, Python has gradually become one of the mainstream computer languages today, and is widely used in artificial intelligence, machine learning and other fields. Using Python language programming, it is convenient to collect, process, organize and analyze various data to build artificial intelligence algorithms to achieve various intelligent applications.

TensorFlow is a deeply open source artificial intelligence learning system developed by Giogle, which has certain advantages over other deep learning frameworks, and its name is expressed as a tensor (T-ensor, n-dimensional array) from a stream (Flow, based on the calculation of a data flow graph One end of the graph flows to the other end of the computing process, supportsPython, C++, Java and other language programming, in the field of image recognition is the first choice of most people, with strong portability and scalability.

2 Dataset Building

200 pepper images were collected in the natural environment, as shown in Figure 1, and the pepper pictures were normalized, manually calibrated, and data divided.

The size of image pixels affects the efficiency of image recognition, reducing the pixel value of the image can improve the network operation speed, this paper scales the collected

Volume 2, Issue 3, March 2023

pepper image pixels to 224224 ×specifications as the input of the network.

FIGURE 1 IMAGES OF COLLECTED PEPPERS

Lab elImg is a target detection image annotation tool, which uses LabelImg to artificially annotate all images after the above processing, and the annotated pictures are saved in an xml document. All the pictures after labeling are randomly divided into training set, verification set and test set according to the ratio of 7:2:1, which is convenient for the later training and testing of the convolutional neural network model, and the data division results are shown in Table 1.

TABLE 1 PEPPER DATA BREAKDOWN

Classification of pepper	Number
Training set	140
Verification set	40
Test set	20

3 CONVOLUTIONAL NEURAL NETWORKS

3.1 ALEXNET MODEL

1In 998, LeEcun et al. proposed the LeNet-5 [13] model, which was widely used in handwritten digit recognition, 2 Alex Krizhevsky, 2012 et al. Alex Net[14] model opened the door to convolutional neural network image recognition,2014 Oxford Visual Geometry Group,2 and Members of Google Deep Mind further developed the VGG[15] (visual geometry group) model for convolutional neural networks. These three models represent three of the development of convolutional neural networks from the rise to the leap

The AlexNet model won the 2012 ImageNet Image Recognition Challenge, which attracted a lot of attention in the image recognition field. Its structure consists of five convolutional layers, three fully connected layers, and three pooling layers alternately with convolutional layers.

The convolutional layer mainly extracts the features of the input feature map through convolution operation. Convolution operation is to use a square convolution kernel, according to the specified step size, slide on the input matrix, through each pixel of the input matrix, the convolution kernel and the corresponding input matrix area elements multiply, sum and add bias terms to obtain a pixel of the output feature, and calculate other pixels in turn. Finally, the activation function is used to nonlinear transform it to obtain a new feature map.

The convolutional layer calculation formula is as follows:

$$C_j = f\left(\sum_{i \in I_i} I_i * W_{ij} + b_j\right) \tag{1}$$

where I_i is the input image; is W_{ij} a convolution kernel; b_j bias term; is a nonlinear $f(\cdot)$ activation function.

The purpose of the pooling layer is to compress the pixels of the feature map, reduce the amount of data that needs to be processed, and speed up the calculation and prevent overfitting problems. There are two commonly used pooling methods, one is Average pooling method, the other is Mass pooling method, Alex Net uses Manax pooling method, in training, pooling is only from the target area to take the maximum value, there are no parameters to learn, and do not change the number of channels before and after the input, It avoids the blurring effect of Average pooling and is robust to small position changes. The Manax pooling operation is shown in Figure 2: in the figure, a convolution kernel with a size of 2×2 and a step of 2 is used to extract the feature map, and the convolution kernel is calculated sequentially until the convolution kernel passes through all regions and pools the node

Bundle.

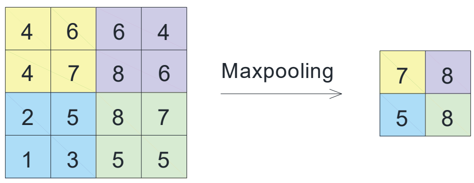


FIGURE 2 POOLING OPERATIONS

The fully connected layer maps the learned features to the sample marker space, integrates the feature representation into a single value, reduces the high dimension to the low dimension, and retains the main information of the image features.

Alex NetmodelDiscard the previously inherited Sigmoid、tbrotherand other functions, optionalReReadfunction as an activation function for the network.ReReadThe function is inx＜0When the gradient is 0, inx＞0The time derivative is 1,This not only simplifies the operation, but also improves the convergence speed of the model, and prevents the problem of

Journal of Research in Multidisciplinary Methods and Applications http://www.satursonpublishing.com/
ISSN: 2957-3920
Volume 2, Issue 3, March 2023

descent and gradient disappearance during the backpropagation processcomputeformula 如 下 Shown,Function image such asFigure 4 shows[16]。

$$f(x)=\begin{cases} x, x>0 \\ 0, x\le 0 \end{cases} \tag{2}$$

After adding D ropout after the fully connected layer, the Dropout operation can randomly discard some neurons in the network temporarily according to a certain probability, reducing the training parameters of the network and greatly preventing the occurrence of overfitting.

3.2 Pepper Recognition Model

Based on AlexNe t model, this paper designs a convolutional neural network model that is conducive to chili recognition based on the advantages of the VGG network model. The 3×3 convolution kernel used in the VGG (visualgeometry group) network structure has improved nonlinear expression ability, which is convenient for extracting more complex features, reducing parameters and reducing the amount of computation. Therefore, this article draws on the advantages of the VGG network to × 11×11 and 5 in the Alex Net network The convolution kernel of 5 is changed to a 3×3 size to minimize parameters and facilitate the extraction of more specific features. The improved model structure and network parameters are shown in Table 2.

3.3 Error Function

Add an error function to the model to calculate the difference between the predicted value of pepper and the actual value (error rate), the size of the error function reflects the strength of the model's fitting ability, and the error backpropagation algorithm is usually used to update the weight of each network layer to minimize the error function result. Cross-entropy error is suitable for classification problems, because this paper studies the identification of peppers, so the cross-entropy error function is selected, and the cross-entropy error function calculation formula is as follows.

$$H=(p,q)=-\sum p(x)\log q(x) \tag{3}$$

4 Experimental Process

4.1 Experimental Environment And Indicators

The operating environment is a 64-bit Win10 system, the processor of the notebook computer is Intel(R) Core(TM) i5-11320H, the main frequency is 3.20GHz, and the memory is 16 GB The training platform is a TensorFlow deep learning open-source framework based on the Python language.

4.2 Training Process

Convolutional neural network has strong self-learning ability, in the training model, you need to pre-set hyperparameters, such as input image size, convolution kernel size, step size, etc., calculate the weight and deviation value parameters of each layer through forward propagation, update the corrected weight through backpropagation, and finally According to the parameters obtained by training, the classification and probability of the image are predicted, and the training process is shown in Figure 3.

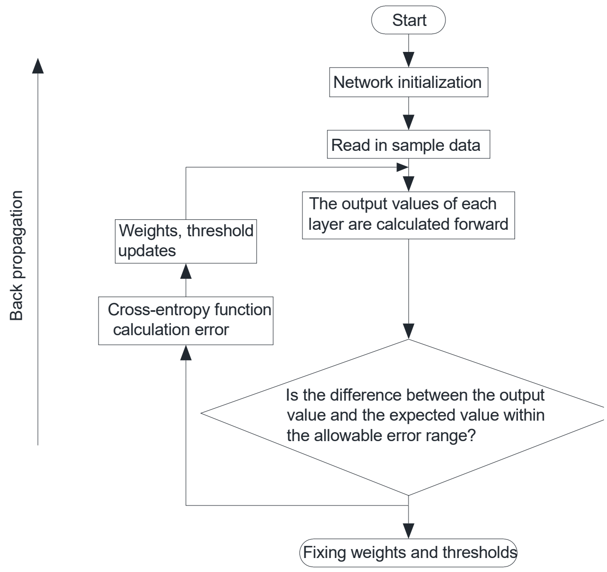
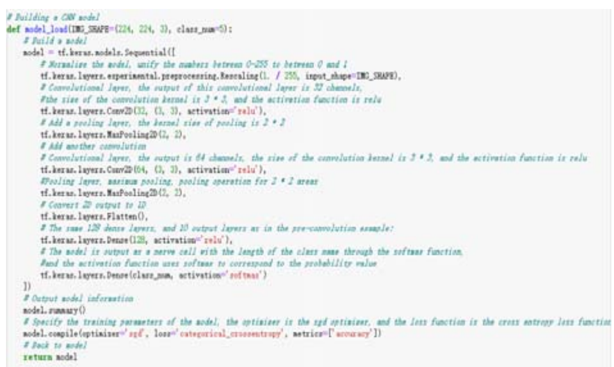


Figure 3: Model Training Flow

The convolutional kernel size of the experimental convolutional neural network is 3×3, and the step size is 1. The pooling method adopts the Mega pooling method; The activation function is the ReLu function; The model is output as a class name length by the softmax function on neurons, and the activation function uses softmax for probability values; The model is optimized by SGD optimizer, and the loss function is the cross-entropy loss function. The convolutional neural network model is shown in Figure 4.



```
# Building a CNN model
def model_load(IMG_SHAPE=(224, 224, 3), class_num=5):
    # Build a model
    model = tf.keras.models.Sequential([
        # Normalize the model, unify the numbers between 0-255 to between 0 and 1
        tf.keras.layers.experimental.preprocessing.Rescaling(1. / 255, input_shape=IMG_SHAPE),
        # Convolutional layer, the output of this convolutional layer is 32 channels,
        #the size of the convolution kernel is 3 * 3, and the activation function is relu
        tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
        # Add a pooling layer, the kernel size of pooling is 2 * 2
        tf.keras.layers.MaxPooling2D(2, 2),
        # Add another convolution
        # Convolutional layer, the output is 64 channels, the size of the convolution kernel is 3 * 3, and the activation function is relu
        tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
        #Pooling layer, maximum pooling, pooling operation for 2 * 2 areas
        tf.keras.layers.MaxPooling2D(2, 2),
        # Convert 2D output to 1D
        tf.keras.layers.Flatten(),
        # The same 128 dense layers, and 10 output layers as in the pre-convolution example:
        tf.keras.layers.Dense(128, activation='relu'),
        # The model is output as a nerve cell with the length of the class name through the softmax function,
        #and the activation function uses softmax to correspond to the probability value
        tf.keras.layers.Dense(class_num, activation='softmax')
    ])
    # Output model information
    model.summary()
    # Specify the training parameters of the model, the optimizer is the sgd optimizer, and the loss function is the cross entropy loss functio
    model.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
    # Back to model
    return model
```

Figure 4: Convolutional Neural Network Model

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN: 2957-3920

Volume 2, Issue 3, March 2023

The result of this model training is not satisfactory, the concept of transfer learning is introduced, the model is normalized, and then the Average pooling method is used for pooling, and the model can be built by outputting the number of classifications through the fully connected layer. The transfer learning model is shown in Figure 5.

```
# Model loading, specifying the size of image processing and whether to perform transfer learning
def model_load(IMG_SHAPE=(224, 224, 3), class_num=5):
    # There is no need for normalization in the process of fine-tuning
    # Load pretrained mobilenet model
    base_model = tf.keras.applications.MobileNetV2(input_shape=IMG_SHAPE,
                                                   include_top=False,
                                                   weights='imagenet')
    # Freeze the backbone parameters of the model
    base_model.trainable = False
    model = tf.keras.models.Sequential([
        # Perform normalized processing
        tf.keras.layers.experimental.preprocessing.Rescaling(1. / 127.5, offset=-1, input_shape=IMG_SHAPE),
        # Setting up the trunk model
        base_model,
        # Globally average pooling the output of the backbone model
        tf.keras.layers.GlobalAveragePooling2D(),
        # Mapped to the final number of categories by the fully connected layer
        tf.keras.layers.Dense(class_num, activation='softmax')
    ])
    model.summary()
    # The optimizer trained by the model is the adam optimizer, and the loss function of the model is the cross entropy loss function
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

Figure 5: Transfer Learning Model

According To The Convolutional Neural Network Model Training Dataset, The Dataset Is Divided Into 5 Classes, Each With 110 Images, Of Which 1 00 Are Used For Training And 10 Are Used For Testing. The Results After Training With The Transfer Learning Model Are Shown In Figure 6.

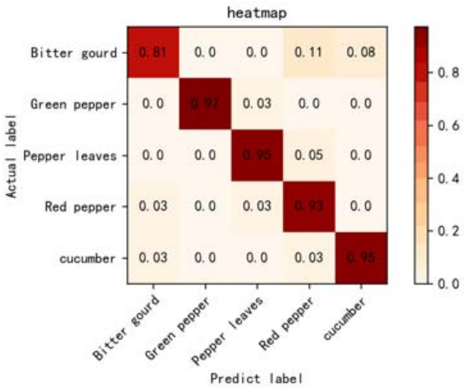


Figure 6: Recognition Accuracy

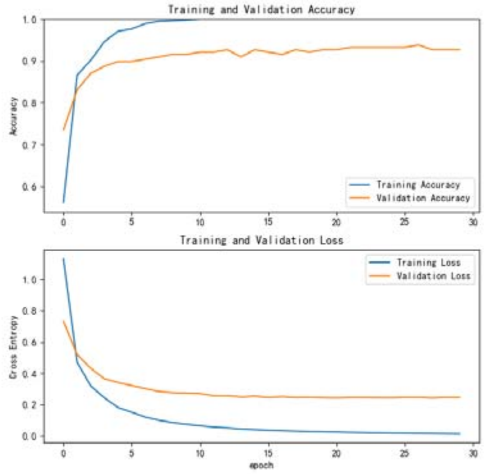


Figure 7 The Loss Function

Cross-entropy is to determine how close the actual output is to the desired output, that is, the smaller the value of the cross-entropy, the closer the two probability distributions, and the cross-entropy loss function trained by the model is shown in Figure 7.

It can be seen from the figure that the correct rate of recognition on the verification set is 90%, and the cross-entropy data is 0. 3。

5 Conclusion

In this paper, the normalization, calibration and data division of the acquired images are firstly analyzed, and then the processing process of the AlexNet model is analyzed, and the VGG model is used to identify the data by the transfer learning method, and the 90% accuracy rate is obtained, and cross-entropy is used Training loss function with a loss value of 0.3.

Funds

College Students Innovation Training Project (202110664030), Zunshi Kehe HZ [2020] No. 10, Zunshi Ke Talent [2021] No. 2, Zunhong Kehe Shizi (2022) No. 06, Zunshi Kehe HZ Zi [2022]. No. 41.

References

[1]XING Zenong, XIN Xin, LIU Yazhong, LI Qiuxiao, ZHANG Jiayuan Current situation and prospect of mechanization of chili pepper industry in China[J]. Bulletin of Agricultural Science and Technology, 2021(06): 229-230+259

[2]SU Anjin,DU Chan,DU Juan. Development and test of self-propelled pepper harvesting machinery[J]. Agricultural Engineering,2021,11(06):17-19

[3]WU Liang. Design and analysis of small walk-behind pepper harvester[D].Chengdu University, 2021. DOI:10.27917/d. cnki.gcxdy.2021.000207.

[4]XU Dongsheng. Design and research of line pepper harvester[D].Gansu Agricultural University,2017.)

[5]FU Longsheng,FENG Yali,ELKAMIL Tola,LIU Zhihao,LI Rui,CUI Yongjie. Image recognition method of multi-cluster kiwifruit in field based on convolutional neural network[J].Transactions of the Chinese Society of Agricultural Engineering,2018,34(02):205-211.)

[6]LI Huaitao,CAO Zhongda,ZHU Chenghui,CHEN Keqiong,WANG Jianping,LIU Xuejing,ZHENG Chengqiang. Cognitive method of intelligent feedback of green plum grade based on deep ensemble learning[J].Transactions of the Chinese Society of Agricultural Engineering,2017,33(23):276-283.)

[7]DUAN Lingfeng,XIONG Xiong,LIU Qian,YANG Wanneng,HUANG Chenglong. Field rice ear segmentation based on deep fully convolutional neural network[J].Transactions of the Chinese Society of Agricultural Engineering,2018,34(12):202-209.)

[8]YANG Yang,ZHANG Yalan,MIAO Wei,ZHANG Tie,CHEN Liqing,HUANG Lili. Research on accurate identification and localization of maize rhizome based on convolutional neural network[J].Transactions of the Chinese Society for Agricultural Machinery,2018,49(10):46-53.)

[9]MA Juncheng, DU Keming, ZHENG Feixiang, ZHANG Lingling, SUN Zhongfu Greenhouse cucumber disease identification system based on convolutional neural network[J] Transactions of the Chinese Society of Agricultural Engineering,2018,34(12):186-192

[10]LIU Deying,WANG Jialiang,LIN Xiangze,CHEN Jing,YU Haiming. Identification method of white-backed planthopper based on convolutional neural network[J].Transactions of the Chinese Society for Agricultural Machinery,2018,49(05):51-56.)

[11]LONG Mansheng,OUYANG Chunjuan,LIU Huan,FU Qing. Image recognition of Camellia oleifera disease based on convolutional neural network and transfer learning[J].Transactions of the Chinese Society of Agricultural Engineering,2018,34(18):194-201.)

[12]YE Jianlong, HU Xinhai Research on image recognition algorithm based on convolutional neural network[J]. Journal of Anyang Normal University,2021(05):14-18.DOI:10.16140/j.cnki.1671-5330. 2021.05. 005

[13]Y. LeCun,L. Bottou,Y. Bengio,P. Haffner. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE,1998,86(11).

[14]Alex Krizhevsky,Ilya Sutskever,Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM,2017,60(6).

[15]Karen Simonyan,Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. [J]. CoRR,2014,abs/1409.1556.

[16]WU Jingzhu,LI Xiaoqi,LIN Long,LIU Cuiling,LIU Zhi,YUAN Yuwei Hyperspectral rapid discrimination of rice origin based on AlexNet convolutional neural network[J]. Journal of Chinese Journal of Food Science,2022,22(01):282-288.DOI:10.16429/j.1009-7848.2022.01.030