Research on pig face recognition model based on keras convolutional neural network

Ke Wang¹, Changxi Chen¹ and Yuxiang He¹*

¹Departments of Computer and Information Engineering, University of Tianjin Agricultural, Tianjin, 300384, China

*Corresponding author’s e-mail: 470799258@qq.com

Abstract: Traditionally, the RFID technology is generally used for the identification of pigs in vivo. However, the method of electronic ear tags and ear tags will cause great pain to the pigs, then ear tags will easily fall off during the pig's activities, increasing the operating cost of the enterprise. This paper uses the powerful feature learning and feature expression capabilities of convolutional neural networks in deep learning to automatically learn the facial features of pigs. Use the Image Data Generator that comes with Keras to perform data enhancement on the pig face pictures of ten pigs and generate pig face dataset. This paper proposes a convolutional neural network model based on LeNet-5 for facial image recognition of pigs. Experimental comparisons were performed by using SGD, Adam and rmsprop optimizers with dropout ratios of 0.3, 0.5 and 0.7. Experiments show that when the SGD optimizer is used and dropout is 0.3, the model recognition rate is the highest, which can reach 97.6%.

1. Introduction

With the rapid development of China’s national economy and the steady improvement of people’s living standards, huge changes have taken place in the diet structure of consumers, and the consumption of meat, eggs and milk has increased significantly [1] [2]. Attention has also increased unprecedentedly. As a daily meat food, pork accounts for a large proportion in China's consumption structure, so the quality and safety of pork is particularly important. At the same time, pigs in a high-density environment, the prevention and control of outbreaks or mass-borne diseases is also very important, but with the intensive and large-scale development of the pig industry, the occurrence of pig diseases has brought great harm to the breeding industry, bring immeasurable economic losses to farmers. At present, there is no complete system for pig identity management and pig identity tracing to establish a data warehouse for the entire growth process of pigs. At present, most of the identity management of pigs use electronic ear tags or ear tags (such as RFID radio frequency technology), and then read the information of the electronic ear tags or ear tags to confirm the identity of the pig. However, electronic ear tags or ear tags can only identify animals at close range, with an accuracy rate of 88.6% [3] and have the following defects:(1) Playing electronic ear tags or ear tags on the ears of pigs will cause physical harm to the pigs; (2) If an electronic ear tag or ear tag is dropped from or destroyed by a pig, then the pig’s identity will be lost; (3) Electronic ear tags or ear tags have a useful life and cannot be reused, increasing the cost of pig farm production management; (4) Ear tags of different pig farms will be duplicated, which is safe for slaughterhouses or pork will cause traceability issues.

For how to accurately identify the individual identity of a pig without harming the pig, face recognition can be used as a reference. Face recognition, as an important research direction in the field
of image recognition, has been successfully used commercially. Face recognition is usually used for non-intrusive access control and monitoring, which is very similar to the application scenario of farming enterprises. Therefore, we can theoretically transfer the related technologies in the field of face recognition to animal identification. At present, there are some related works at home and abroad that have applied face recognition technology to pig behavior recognition [4], and have achieved very good results in sheep and dog identification [5,6]. However, due to the short growth cycle of pigs, rapid changes in appearance, and difficulty in identification, the inbreeding characteristics of pigs will lead to a high degree of facial similarity among individuals. At the same time, in some pig houses, pig faces that have not been cleaned for a long period of time will also hide their facial features, which makes it difficult to recognize pig faces.

In recent years, deep learning has rapidly increased in the academic community. An important basis for supporting deep learning is that the human brain has a very rich hierarchical structure. In 2012, AlexNet [7]’s successful performance in the ImageNet competition directly established the important position of CNN. Later, deeper network models such as VGG [8] and GoogLeNet [9] have appeared for image classification and recognition. Taigman team proposes DeepFace for face recognition [10]. Among domestic research scholars, Tang Xiaoou's team at the Chinese University of Hong Kong proposed DeepID for face authentication [11]. In terms of using animal facial features for individual recognition, Emmanuel Okafor's team abroad uses convolutional neural networks for individual identification of wild animals [12]. Freytag’s team uses convolutional neural networks for individual identification of African chimpanzees [13]. Because the structure of the convolutional neural network is non-linear, this makes it a good solution to some non-linear factors in facial recognition. For animal facial recognition based on the convolutional neural network, further research is needed.

There are also some similar competitions that have promoted the development of animal facial recognition, such as the Jingdong Pig Face Recognition Competition in 2017, and some animal recognition applications, such as Microsoft's what-dog.net, but some applications such as what-dog.net are mostly coarse-grained recognition, which only recognizes the category of animals, but fails to perform finer-grained recognition, such as identifying different individuals in the same category.

On the basis of collecting a large number of pig face images, this paper uses CNN to recognize pig face pictures under complex backgrounds to avoid human subjective factors affecting the recognition results. According to the characteristics of pig face images in the pig house environment, LeNet-5 [14] convolutional neural network was optimized, different optimizers and dropout ratios were used to establish a pig face image recognition model based on convolutional neural network. Realize fast and effective recognition of pig faces in complex environments.

2. Material and Methods

2.1 Keras profile

Keras is a deep learning library that abstracts neural networks at a high level and encapsulates rich and friendly APIs. Keras is written in Python, with Tensorflow, Theano or CNTK as the back end. Keras itself is used as the front-end for writing neural networks, which is equivalent to packaging another layer on the basis of libraries such as Tensorflow, you can freely switch between Tensorflow, Theano and CNTK below it. For the users, there is basically no need to modify the code or only a small part of the code. It can be said that it is an intermediate layer that encapsulates the Tensorflow, Theano and CNTK APIs. Because Keras supports multiple back-end engines, Keras does not lock users into an ecosystem and facilitates migration. At the same time, Keras can more easily turn models into products.

Keras is widely adopted by industry and academia, is the most used deep learning framework besides Tensorflow in industry and academia. At present, the development of Keras is mainly supported by Google, and the Keras API has been wrapped in Tensorflow. Keras was born for rapid experimentation and enables quick realization of ideas. It has the following characteristics: (1) Supports simple and fast prototype design, is user-friendly, highly modular, and has good scalability.
(2) Support convolutional neural network and recurrent neural network at the same time, and the combination of both. (3) Seamless switching between CPU and GPU.

2.2 Convolutional neural network
Convolutional neural network [15] is a relatively popular deep network. Unlike the traditional network structure, it contains very special convolutional layers and pooling layers, where the convolutional layer and the previous layer are locally connected and pooled. The weights are connected in a shared manner, which greatly reduces the number of parameters. The pooling layer can greatly reduce the input dimensions, thereby reducing the complexity of the network, making the network more robust, and effectively preventing overfitting. Due to the above design, the convolutional network is mainly used to identify two-dimensional graphics that are invariant to scaling, displacement, and other forms of distortion, and can directly take the original picture as input without the need for complicated preprocessing.

2.3 LeNet-5 neural network
The LeNet-5 algorithm is a convolutional neural network proposed by Y. LeCun. It is a special multilayer neural network. Like other neural networks, it is also trained by back propagation. The difference is its network structure. Its biggest feature is the weight sharing between each layer, which allows LeNet-5 to reduce a large number of parameters in the process of building a network and speed up the learning process. Its network structure is shown in Figure 1.

Figure 1. Network structure of Lenet-5

The convolutional neural network LeNet-5 does not include inputs, and consists of 7 layers, each of which includes trainable parameters (weights). The input of the network is a 32 × 32 image, of which the C layer is composed of the convolutional layer neural. The network layer is composed of meta-elements, and the S-layer is a network layer composed of sub-pooling layer neurons.

The network layer C1 is a convolutional layer composed of 6 feature maps. Each neuron is connected to a 5 × 5 neighborhood of the input image, so the size of each feature map is 28 × 28.

The network layer S2 is a sub-pooling layer composed of six feature maps with a size of 14 × 14. It is obtained by sampling the C1 layer. Each neuron of the feature map is connected to a 2 × 2 neighborhood of the C1 layer.

Network layer C3 is a convolutional layer composed of 16 feature maps of size 10 × 10. Each neuron of the feature map is connected to a 5 × 5 neighborhood of several feature maps of the S2 network layer.

The network layer S4 is a sub-pooling layer composed of 16 feature maps with a size of 5 × 5. Each neuron of the feature map is connected to a 2 × 2 size neighborhood of the C3 layer.

Network layer C5 is a convolutional layer composed of 120 feature maps. Each neuron is connected to a 5 × 5 size neighborhood of all feature maps of the S4 network layer.

Network layer F6, including 84 neurons, is fully connected to network layer C5. Finally, there are 10 neurons in the output layer, which are composed of radial basis function units (RBF). Each neuron in the output layer corresponds to a character category.

2.4 Improved LeNet-5
If the LeNet structure is directly introduced into the feature extraction and classification of pig face
images, considering the differences from the samples (handwritten characters) used in the original network and the imaging channel of the pig face images, this paper uses 224x224x3 images as the model input. Since the pig face image is less affected by factors such as torsion and deformation, we can reduce the number of local receptive fields in each convolutional layer in the original LeNet network to improve the training speed of the network. After training neural networks with different structures, by comparing the accuracy and training time of different model recognition, the size of the local receptive fields used in the end is 5 × 5, the number of local receptive fields of the three convolutional layers is 6, 16, and 32. The pooling windows are all 2 × 2 in size. The number of fully connected layer neurons was 120, 84 and 10. The network structure of the model based on Keras can be obtained as shown in Figure 2.

| Layer (type)         | Output Shape          | Params # |
|----------------------|-----------------------|----------|
| conv2d_1 (Conv2D)    | (None, 220, 220, 6)   | 456      |
| max_pooling2d_1 (MaxPooling2) | (None, 110, 110, 6)  | 0        |
| conv2d_2 (Conv2D)    | (None, 106, 106, 16)  | 2416     |
| max_pooling2d_2 (MaxPooling2) | (None, 53, 53, 16)  | 0        |
| conv2d_3 (Conv2D)    | (None, 49, 49, 32)    | 12832    |
| max_pooling2d_3 (MaxPooling2) | (None, 24, 24, 32) | 0        |
| flatten_1 (Flatten)  | (None, 18432)         | 0        |
| dense_1 (Dense)      | (None, 120)           | 2211960  |
| dropout_1 (Dropout)  | (None, 120)           | 0        |
| dense_2 (Dense)      | (None, 84)            | 10164    |
| dense_3 (Dense)      | (None, 10)            | 850      |

Total params: 2,238,678  
Trainable params: 2,238,678  
Non-trainable params: 0

Figure 2. Network structure of the model

During the training process, for the fully connected layer, due to the huge amount of parameters, overfitting is easy to occur [16]. We introduce the Dropout layer after the fully connected layer as shown in Figure 3. The dropout layer was proposed by the Hinton team [17], during the deep learning network training process, the neural network units of the fully connected layer are temporarily discarded from the network with a certain probability. Because it is randomly discarded, each batch is training a different neural network, which increases the robustness of the network and reduces the occurrence of overfitting.
Figure 3 Schematic diagram of the Dropout layer

The activation function uses rectified linear units (ReLU), which enables the model to better mine related features without causing gradient dispersion problems like the Sigmoid function. As Max-pooling is a non-linear down-sampling method, it can reduce the error of feature extraction caused by the deviation of the estimated mean caused by the error of the convolution layer parameters to a certain extent. Max-pooling was selected as the down-sampling method in the experiment. The output layer uses Softmax as the activation function. Softmax function expressions are as follows:

\[ S_i = \frac{e^{V_i}}{\sum_{j=1}^{C} e^{V_j}} \]  

Where \( S_i \) is the output of the classifier's previous network, \( i \) represents the category index, and the total number of categories is \( C \), which is the ratio of the index of the current element to the sum of the indices of all elements.

When we choose Softmax as the activation function of the output layer in the establishment of the multi-classification model, and the loss function used by it is generally a cross-entropy loss function, the cross-entropy loss can be written as:

\[ L = \frac{1}{N} \sum_i L_i = \frac{1}{N} \sum_i -\log\left(\frac{e^{V_i}}{\sum_{j=1}^{C} e^{V_j}}\right) \]  

Where \( N \) is the number of categories in the system, \( L_i \) is the output of the classifier's pre-stage network, and \( i \) is the category index.

2.5 Data preprocessing

2.5.1 Data acquisition

This article studies the video files provided by the pig companies. Each file corresponds to an individual Changbai pig in a pig house scene, and the duration is about 1 min. Shoot in natural conditions, capture videos of pigs in different angles and light at different angles, and get the posture and expression of pigs in different environments. The pictures taken from multiple angles are used to improve the generalization ability of subsequent training models. Also during the collection process, we did not deliberately ask the pig's face to be clean, so as to obtain a clean and dirty pig face pictures, the purpose is to extract more diverse facial features in the feature extraction process. The specific picture is shown in Figure 4. On the basis of ensuring that each image contains only one individual pig, 197 frame images were intercepted from each video, and a total of 1970 initial dataset images of 544 × 411 pixels were obtained for 10 pigs.
2.5.2 Data enhancement

In order to improve the classification accuracy, deep convolutional neural networks require a large number of training samples. Randomly perform image enhancement operations such as rotation, translation, hue, brightness, shading, scaling and adding salt and pepper noise disturbance to 600 cropped $224 \times 224$ pixel images to expand data samples. The operations such as the shading area and the zoom ratio are randomly selected to ensure the randomness of the generated image.

The data enhancement in this paper is implemented using the Image Data Generator that comes with Keras. The pig face picture after data enhancement is shown in Figure 5. The Image Data Generator class mainly generates new pictures that have been changed by means of generators. The Image Data Generator parameters are shown in Table 1.
Table 1. ImageDataGenerator parameters

| parameter           | Value |
|---------------------|-------|
| width_shift_range   | 0.2   |
| height_shift_range  | 0.2   |
| shear_range         | 0.2   |
| rotation_range      | 40    |
| horizontal_flip     | True  |
| fill_mode           | nearest |
| zoom_range          | 0.2   |

Height_shift_range and width_shift_range in the table are the vertical and horizontal position shift values. Shear_range indicates the picture shearing strength, rotation_range indicates the degree of random rotation, horizontal_flip indicates whether to flip horizontally at random, and fill_mode indicates that when the transformation is performed, the points that exceed the boundary will be based on this. The method given by the parameter is processed, and zoom_range represents a random zoom range.

After the above preprocessing operation, the entire data set contains a total of 1000 224 × 224 pixel pictures, which are used for model input. Finally, the data set is divided into a training set and a validation set at a ratio of 8:2. In addition, although the above preprocessing operation is performed on the Changbai pig data set, the pig breed information was not deliberately considered during the entire operation, so it is also applicable to the individual pig pretreatment of breeds such as black-haired pigs and Ningxiang pig process.

2.6 Pig face recognition based on convolutional neural network

Related modules of Keras are imported, and the improved LeNet-5 model is constructed based on Keras. The network model diagram is shown in Figure 6.

![Figure 6. Improved LeNet-5 convolutional neural network model](image)

The model adopts the LeNet-5 model structure, which is a stack of 3 convolution modules and 3 fully connected modules. The output number of the last fully connected layer is 10, corresponding to the number of target categories. SoftMax is used to calculate the loss, as shown in Figure 6. The convolution module is composed of a convolution layer and a maximum pooling layer. The fully connected module F7 contains a dropout layer, and some neurons are randomly dropped during training. We use 0.3, 0.5, and 0.7 to compare the drop probability, and select the drop probability with the best effect as the final parameter of the model. Pooled down sampling maintains a certain translation invariance, and dropout reduces overfitting. The activation function of the neuron is a linear rectified function (ReLU). Its unilateral suppression characteristics make the neuron have sparse
activation, which can effectively solve gradient dispersion and help accelerate network convergence.

2.7 Model Optimization
The essence of network model optimization is the process of iteratively minimizing the loss function. In this paper, stochastic gradient descent (SGD), rmsprop and Adam are used to optimize the model, and the optimizer with the best effect is selected by comparison. The learning rate update strategy uses an exponential decay method. In the initial stage of training, a large learning rate is set to quickly reach the vicinity of the optimal solution, and then the learning rate is gradually reduced to avoid fierce oscillations due to the large learning rate. Finally, cross-entropy is used to calculate the classification loss.

3. Results
The neural network learning framework uses the Keras framework. The improved LeNet-5 neural network proposed above is used for feature extraction and training on the pre-processed pictures of the pig face dataset. The training epoch is set to 200 and the batch batch size is 40. Take Adam optimizer, Dropout is 0.3 as an example, the training set reaches 199 steps, the accuracy rate is 97.13%, and the loss function loss value is 0.0934. The test set accuracy at this time was 89.5%, and the loss value was 0.8343. The specific iterative process is shown in Figure 7.

![Figure 7. Model training process](image)

The accuracy and loss rate of the model training process on the pig face data set changes after training, we use Python visualization tool matplotlib to get the accuracy and loss rate change curves, as shown in Figure 8.
4. Discussion

The difference in accuracy of different optimizers and different Dropout ratios during the iteration is shown in Table 2. Through comparative experiments, we can get several conclusions about the influence of the optimizer and Dropout on the model: When the Dropout ratio is 0.7, the model recognition effect is the worst, and the ratio of 0.3 is the best. Iteration results of the SGD optimizer under dropout are the best.

| Dropout | 0.3 | 0.5 | 0.7 |
|---------|-----|-----|-----|
| SGD     | 97.6% | 95.5% | 85.0% |
| Adam    | 97.1% | 86.7% | 73.3% |
| Rmsprop | 96.6% | 91.2% | 68.5% |

5. Conclusion

This paper introduces a CNN-based pig face image recognition method, which optimizes the parameters and structure of LeNet-5 neural network models. In the pig face dataset training, Keras is used to compare the model training process under different optimizers and Dropout ratios. Experiments show that under the SGD optimizer, the picture recognition rate is the highest when the Dropout ratio is 0.3, and the recognition rate reaches 97.6%.

At present, the model used in this paper can accurately identify the pig face categories in the data set, but the model structure is relatively simple. When applied in practice, it is also necessary to increase the types and number of learning samples, deepen the model network structure, and further improve the generalization ability of the model.

Acknowledgements

This research was financially supported by Mr. He. Her serious scientific attitude, rigorous academic spirit and excellence in her work style deeply infected and inspired me. Mr. He not only gave me careful guidance in my studies, but also gave me meticulous care in thought and life. Here, I would like to extend my sincere gratitude and high respect to Mr. He. I also want to thank the group of students who have spent a happy research life together. Thanks to their help and support, I was able to overcome the difficulties and doubts one by one until the completion of this article.

Finally, I would like to express my heartfelt thanks again to the teachers and classmates who cared for and helped me!
Fund Project: Major project of Internet integration and innovation plan of Tianjin Science and Technology Bureau, R & D and Application of Livestock and Poultry Precision Production Based on Internet of Things and Big Data Analysis(18ZXRHNC00080)

References

[1] Tao W, Liu Y, Zhang J, Liu Y. Pork Consumption Situation in China in Recent Years and Prospects for the “13th Five-Year Plan” [J]. Agricultural outlook, 2017, 13(09): 108-113.

[2] Tong L, Liu Y, Pan C. Pork Consumption Characteristics and Changes in Chinese Residents [J]. China Animal Husbandry, 2015(20): 41-43.

[3] J. Maselyne, W. Saeyes, B. De Ketelaere, K. Mertens, J. Vangeye, E.F. Hessel, S. Millet, A. Van Nuffel, Validation of a High Frequency Radio Frequency Identification (HF RFID) system for registering feeding patterns of growing-finishing pigs, Comput. Electron. Agric. 102 (March (Supplement C)) (2014) 10-18.

[4] Ahrendt P, Gregersen T, Karstoft H. Development of a real-time computer vision system for tracking loose-housed pigs[J]. Computers and Electronics in Agriculture, 2011, 76(2): 169-174.

[5] U. Gonzales Barron, G. Corkery, B. Barry, F. Butler, K. McDonnell, S. Ward, Assessment of retinal recognition technology as a biometric method for sheep identification, Comput. Electron. Agric. 60 (March (2)) (2008) 156-166.

[6] S. Kumar, S.K. Singh, Monitoring of pet animal in smart cities using animal biometrics, Future Gener. Comput. Syst. (December) (2016).

[7] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097-1105, 2012.

[8] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in neural information processing systems. 2012: 1097-1105.

[9] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1-9.

[10] Taigman Y, Yang M, Ranzato M A, et al. Deepface: Closing the gap to human-level performance in face verification[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1701-1708.

[11] Sun Y, Wang X, Tang X. Deep Learning face representation from predicting 10000 classes[C]. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1891-1898.

[12] Okafor E, Pawara P, Karaaba F, et al. Comparative Study Between Deep Learning and Bag of Visual Words for Wild-Animal Recognition[J].

[13] Freytag A, Rodner E, Simon M, et al. Chimpanzee faces in the wild: Log-euclidean cnns for predicting identities and attributes of primates[C]. German Conference on Pattern Recognition. Springer International Publishing, 2016: 51-63.

[14] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to hand written zipcode recognition[J]. Neural computation, 1989, 1(4) : 541-551.

[15] Lawrence S, Giles C L, Tsoi A C, et al. Face recognition: a convolutional neural-network approach[J]. IEEE Transactions on Neural Networks, 1997, 8(1):98-113.

[16] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. 2012: 1097-1105.

[17] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.