ISSN: 2664-9640

Design and Implementation of Animation Recognition System

Jinyu Sun* , Shilong Yang

School of Huaiyin Normal University, Huaian 223001, China;

Abstract

With the increasing demand for entertainment such as animation and games, various games and animation characters have increased. It is time-consuming and laborious to identify specific animation characters manually. In this paper, the system is based on the Python, PyQt5 graphics framework and uses the TensorFlow machine learning framework combined with transfer learning to build a MonaCNN recognition model for image recognition and screening. The experimental results show that the constructed neural network model has an absolute accuracy of 95.1%, which solves the problem that users want to pick out pictures with specific characters from the pictures and save them, and at the same time, exclude pictures without corresponding characters in the picture.

Keywords

Python, tensorFlow, image identification, neural networks.

1. Introduction

The advancement of science and technology has enabled the development of smartphones and the internet to provide people with a large amount of picture information. Because pictures are not restricted by factors such as geography and language, they have the ability to replace complex and cumbersome words and have become an essential means of conveying information today. Such as facial expressions and emoticons commonly used by people in chat software. However, as the primary information carrier on the internet has changed to pictures, problems have arisen one after another [2]. When the information is recorded in words, people can easily find the desired content and edit it arbitrarily through keyword searches, but when the information is recorded in the form of pictures, the content in the pictures cannot be retrieved, which affects people from the pictures—efficiency in finding key content. Fast information recording and sharing methods are inseparable from pictures, but it does cause many obstacles to the efficiency of information retrieval.

Image recognition technology is one of the critical technologies in the field of computer vision. It is a technology for computers to process, analyze and understand images to identify various patterns of targets and objects [4]. The recognition process includes image preprocessing, segmentation, feature extraction, and judgment matching. For simplicity, image recognition lets the computer read and understand the content of the image by viewing and analyzing the image like a human. Image recognition technology is widely used in face recognition, autonomous driving, agricultural disease detection, and other fields [5]. With the advancement of current technology, deep learning to train neural network models for image recognition has become a significant mainstream. Since the AlexNet [6] network was proposed in 2012, the deep neural network has occupied the ImageNet list [7].

In order to further optimize image recognition technology, transfer learning [8] is also an efficient technology. It refers to a pre-trained model being reused in another task. This kind of transfer, known as inductive transfer in deep learning, narrows the search of possible models in an advantageous way by using a model that is suitable for a different but related task. With the help of this learning, we can improve the accuracy of the model.

ISSN: 2664-9640

The purpose and significance of designing the system in this paper are to use the current technology [9], [10] to help animation fans identify specific animation characters faster and more safely and to screen and classify pictures. Compared with other image recognition programs, this system uses the TensorFlow framework [11] and combined with transfer learning, a neural network model that can be run can be quickly built. In addition, the GPU is used to accelerate the operation, which saves the construction speed of the neural network model, which is convenient for users to reduce the time cost of building the neural network model.

2. Model Design

The main function of this system is to help users quickly and intelligently identify pictures containing specific anime characters from a large number of random pictures (this system takes the female character Mona in the game Genshin Impact as an example). In order to achieve better recognition accuracy, the model's training process is divided into two parts: pre-training and transfer learning. First, the training set is adjusted to two types of male characters and female characters in the game Genshin Impact. After the first training, a neural network model that is more sensitive to gender characteristics is obtained, and the model obtained this time is used as a pre-training model. Based on the pre-trained model, the training set is adjusted into two categories: clothing, body shape, and hair color that are similar and dissimilar to Mona, and then transferred learning to obtain a model that is sensitive to characters with physical characteristics similar to Mona. Finally, the training data set is adjusted into two categories: Mona and Others, and then a round of transfer learning is performed to obtain a neural network model that can better identify the character Mona, that is, the MonaCNN model of this system.

The neural network model building part mainly realizes the building of the neural network model and improves the accuracy of model recognition through pre-training and transfer learning. It is also possible to re-select the training sample dataset and then re-train MonaCNN to complete the classification of different anime characters or other content.

The neural network model MonaCNN in this paper, and its network hierarchy is shown in Figure 1.

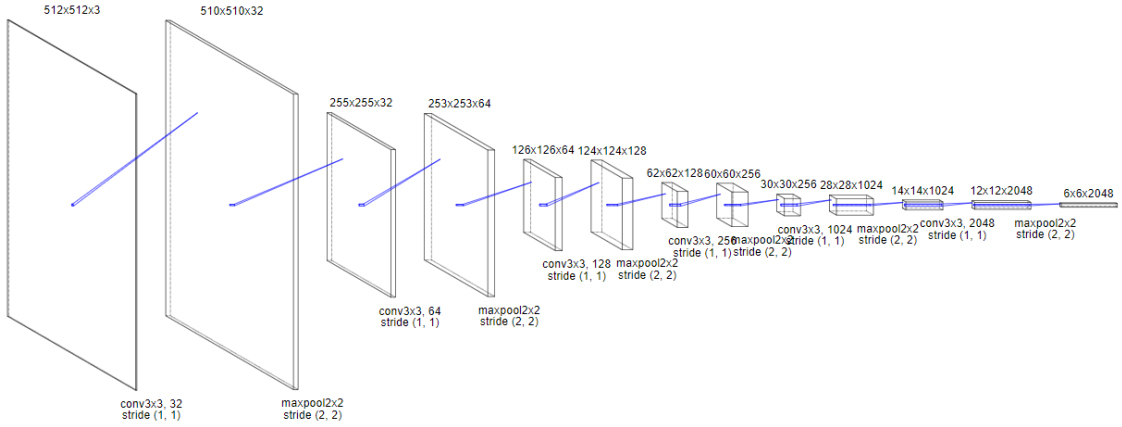

Figure 1: Neural Network Hierarchy

The preprocessing layer will format the image to a size of 512×512 and, at the same time, map the RGB value of the image from [0, 255] to the [0, 1] interval for grayscale processing. Seven sets of convolutional pooling layers follow the preprocessing layer. The number of neurons is 3, 64, 128, 256, 512, 1024, and 2048 respectively; the convolution kernel size is 3×3; the activation function is ReLU. After this process, the image is convolved from 512×512×3 to 2×2×2048. Then connect two fully connected layers with Dropout, the number of neurons is

ISSN: 2664-9640

2048 and 1024 respectively, the Dropout parameter is 0.3; the activation function is ReLU. Finally, the output layer is connected. The output layer has two neurons, which output the probability values Pa and Pb of the output images in two categories. Bring Pa into formula (1) to determine the classification status of the input image.

$$f(x)=\begin{cases}0, & P_A \geq P\\ 1, & P_A < P\end{cases} \tag{1}$$

where P is the user-defined minimum confidence, and the default is 95%. When $f(x)=0$ the picture is considered to belong to the category containing Mona, when $f(x)=1$ the picture is considered to belong to the category of there is no Mona in the picture. For the neural network construction, the model training flow chart is shown in Figure 2, and the main program is saved in the Load_Model.py file.

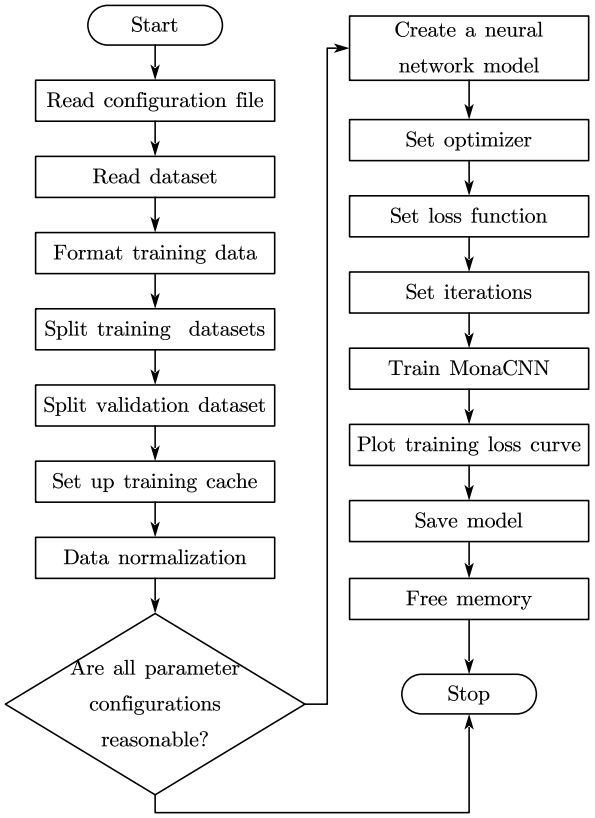

Figure 2:Neural network model training flow chart

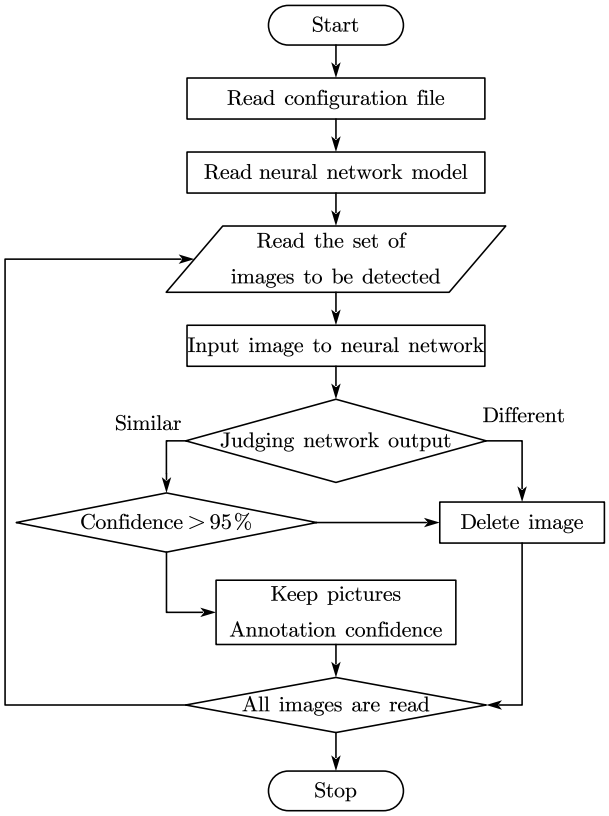

Figure 3: Recognition program work flow chart

First, the program will read the pre-set network parameters, such as the resolution of the input image, the number of samples selected for one training, and so on. Second, the program will read the image dataset and convert all the read images to the exact resolution. Again, the program splits the training data set, some of the data is used to train the neural network, and some is used to test the training results. Next, the program will automatically set the appropriate training cache according to the computer's graphics card hardware and perform data standardization and expansion. So far, if the parameter configuration is reasonable, the imported data can be trained. Otherwise, the program will be stopped.

At this point, the program will create a neural network model, read the optimizer, loss function, and iteration number set in the program and start training the model. After the training is completed, the program will draw the training loss curve to represent the model's training process and final accuracy. Finally, the program saves the model and frees the memory used. At this point, the trained model can be used to classify the existing images.

ISSN: 2664-9640

For the recognition module of the model, the recognition program workflow chart3 is shown. First, the recognition program will read the configuration file to complete the relevant settings for the program, such as the position of the image to be detected and the size of the confidence level. Then read the neural network model trained by the Save_Model.py file, and input the pictures to be detected in turn for judgment. The image will be retained when the detected image has a greater probability of belonging to the target class, and the confidence level is greater than 95%. When the result of the system's image target detection is greater than 95%, the image belongs to the mona class; when the detected image does not belong to the mona class, Pictures are removed when the probability of the target class is greater. The program will end after all the images to be detected been recognized. The main program is saved in the Save_Model.py file.

Figure 4: Mona

3. Dataset

The samples involved in training the MonaCNN model in this paper are self-built datasets named Mona. The main content of the dataset used for identification, Mona, is the female anime character wearing a witch hat and dark blue to black hair in Figure 4. Because this model mainly performs two-class recognition, the data set samples are divided into mona and other categories. For the collection of samples in this dataset, firstly, 300 images were downloaded in batches from the gallery website *wallheaven* with Genshin impact as the keyword as the initial sample; secondly, the corresponding mona and others were placed in the corresponding mona and others according to whether the character content in the images contained Mona. These two folders; then, according to model training needs, the dataset's number of samples is increased to 3000 images. When building the model, the program will automatically divide the 3000 images into a training set and a test set in a ratio of 8:2 for training and testing.

4. Experiments

In the initial test, it was found that the predictive ability of the first generation of MonaCNN was not ideal. The model focused too much on the clothing color and hair color of the characters in the picture, which caused it to mistake characters with similar clothing and hair color as recognition targets. Therefore, in order to improve the recognition accuracy of the MonaCNN neural network model, the training method of transfer learning is selected. First select characters whose clothing and hair color are similar to Mona as one category and other characters whose clothing and hair color are different from Mona as another category.

After the pre-training is completed, the normal sample set (the picture with Mona in the picture; the picture without Mona in the picture) is trained to obtain a model that does not pay too much

ISSN: 2664-9640

attention to the character's clothing and hair color. The training loss curve in Figure 5 shows the model training process and final accuracy.

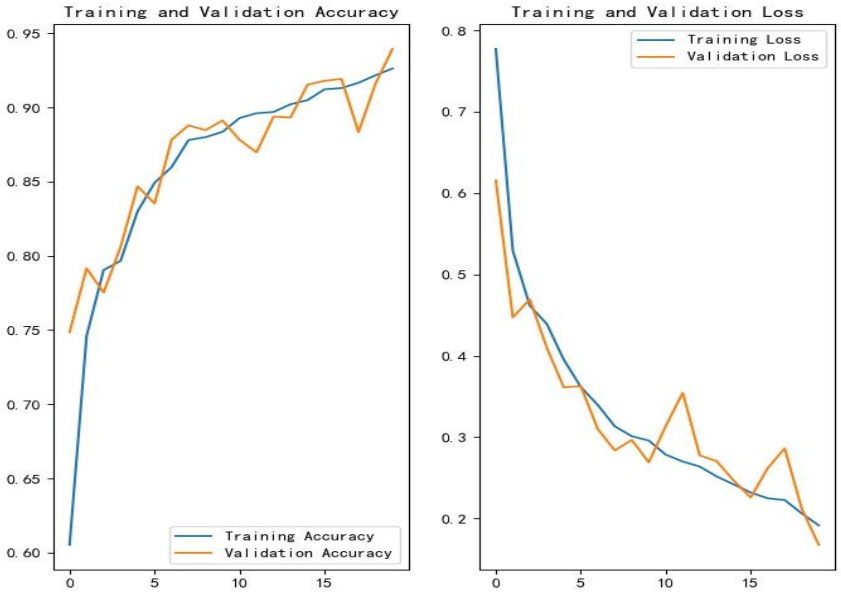

Figure 5: Training loss curve

The functions of the four curves in the figure are all used to measure the quality of the prediction of the neural network model0. That shows the gap between the prediction and the actual data. The greater the degree of fluctuations, the higher the volatility predicted by the model. The blue curve in the left figure represents the training accuracy, and as it rises, the accuracy of the model predictions gradually increases. The orange curve represents the validation accuracy, which represents the accuracy of the validation data. The blue curve in the figure on the right represents the training loss, and when it decreases, it also indicates that the model's accurate prediction is increasing. The orange curve represents validation loss, which is the opposite of validation accuracy. Figure 5 shows the training loss curve of the current optimal neural network model.

The change of the loss curve shows that the fluctuation of the neural network model is gradually stabilized, and the curve is gradually smoothed after multiple pieces of training. The absolute recognition rate for the recognition target Mona (with 2268 pictures as the test data set) has gradually increased from 83.51% of the first generation model to 96.3% of the currently used model. The training comparison is as follows:

For the first-generation MonaCNN and the final MonaCNN used in this system, the same 2268 test images are used, and the test results are shown in Tables 1 and 2. The first-generation MonaCNN has only an absolute accuracy of 83.51% for the recognition target. After several pieces of training, the accuracy of MonaCNN on the recognition target has reached 96.3%, and it has been able to complete basic recognition tasks with high accuracy. Therefore, the final MonaCNN is used as the neural recognition network of this system.

Table 1: First-generation model test results

Classification	Data
Total number of images	2268
Total number of actual Mona classes	156
The total number of Mona classes classified by the model	420
Total number of unidentified Mona	55
Misidentified as Mona Total	319

ISSN: 2664-9640

Number of misclassifications	374
Mona single class accuracy	64.74%
Absolute accuracy	83.51%

Table 2: Final model test results

Classification	Data
Total number of images	2268
Total number of actual Mona classes	156
The total number of Mona classes classified by the model	188
Total number of unidentified Mona	25
Misidentified as Mona Total	57
Number of misclassifications	82
Mona single class accuracy	83.97%
Absolute accuracy	96.3%

In order to test the recognition ability of the current MonaCNN model, 3000 pictures with the keyword Genshin impact were randomly downloaded from the picture website. These 3000 images are pre-classified by manual recognition and used as a test data set to test the usage and classification accuracy of the system in general use environments. For these 3000 test images' results are shown in Table 3.

Table 3: Test results

Classification	Data
Total number of images	3000
Number of images with Mona	463
The number of Mona pictures classified by the model	449
Unrecognized pictures of Mona	79
Number of images incorrectly identified as Mona	68
The number of overall misclassifications	147
Mona single class accuracy	83%
Absolute accuracy	95.1%

Among them, the time consumed by the prediction and classification of a single image is: 0.06~0.27 seconds, the average is 0.075 seconds, and the mode is 0.06 seconds.

5. System design

For convenience, this system uses PyQt5 to design the user interface. As shown in Figure 6, the system consists of three parts: the main interface, the setting interface, and the identification interface. The function of the main interface is to guide the user to enter the setting interface or enter the identification interface. Otherwise, exit the program directly. The user can click the setting interface button to enter the setting interface, click the identification interface button to enter the identification interface or click the exit program button to end the running program.

After the user clicks the setting interface button, the user will come to the setting interface, where the user can select the image compression file path to be recognized and set the confidence level of the image (the range is between 0-1, the larger the value, the higher

ISSN: 2664-9640

precision is required). You can also set whether to copy the recognized pictures and choose the storage path. After completing the setting, you can click the Confirm Setting button to save the setting or click the Return to return to the main page.

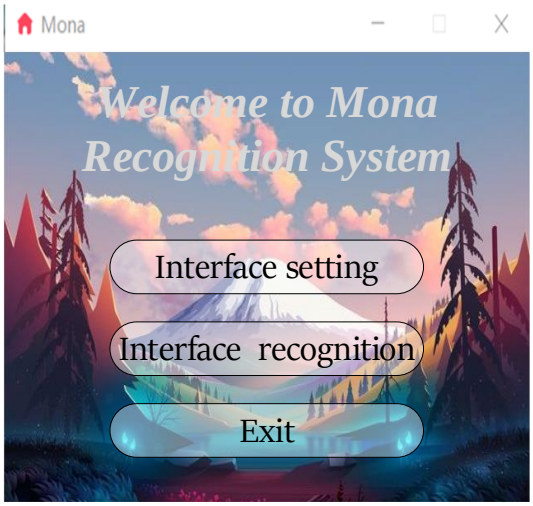

Figure 6: System interface

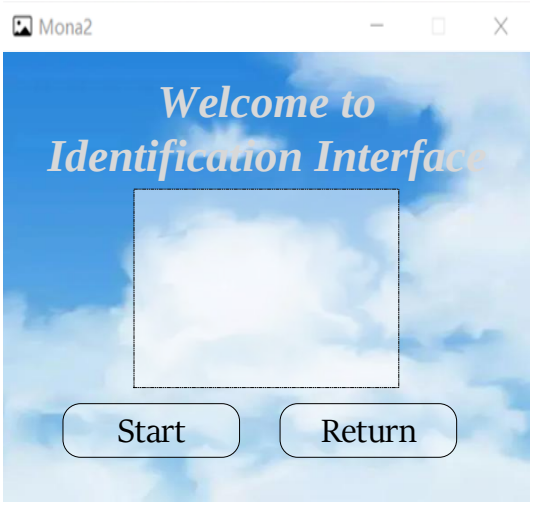

Figure 7: Identification interface

After the user clicks the identification interface button, the user will come to the identification interface. As shown in Figure 7, the user can click the Start Recognition button on the current page to execute the recognition procedure, and the system will make corresponding changes to the relevant steps in the recognition procedure according to the settings of the user in the setting interface. The function of the recognition program is to send the batches of pictures saved on the local hard disk to the neural network for picture content detection. According to the output result of the neural network, pick out the pictures that contain the anime character Mona from the pictures saved in batches on the local hard disk. Some pictures are saved, and others that do not have Mona in the picture are deleted. After the recognition is completed, the program will display the recognition result in the text box, and the user can intuitively understand the various information of the recognized picture or return to the main interface to perform other operations.

6. Conclusion

The system designed in this paper is mainly aimed at the complex problem of recognition of specific image targets. Computer image recognition technology is used instead of manual recognition, and a Python-based animation character recognition system is designed to solve the problem of wasting a lot of manpower and time by relying on manual recognition. The system is developed using Python language and the TensorFlow machine learning framework. It can be used on all PCs with complete programs and corresponding settings. It can complete the essential recognition of the primary recognition target Mona. In the face of 3000 random pictures, An absolute accuracy of 95.1% was achieved in the test. The system can also change the default recognition target of the program to other recognition targets by replacing the training data set of the neural network, which has specific practical value.

References

[1] Chelali F. Z, Djeradi A, Pérez-Cisneros M. Face Recognition Using MLP and RBF Neural Network with Gabor and Discrete Wavelet Transform Characterization: A Comparative Study[J]. Mathematical Problems in Engineering, 2015, 56(25): 1025-1036.

[2] Sisodia D, Singh L, Sisodia S. Fast and Accurate Face Recognition Using SVM and DCT[M]. Springer India: 2014.

ISSN: 2664-9640

[3] Ahonenbinary T, Hadid A, Pietikainen M. Description with local patterns recognition[J]. Transactions on application to Pattern Analysis Machine Intelligence, 2014, 28(12): 2037-2041.

[4] Taigman Y, Ming Y, Marc'Aurelio Ranzato, 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.

[5] Szegedy C. Going deeper with convolutions[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1-9.

[6] Angshul M, Richa S, Mayank V. Face Verification via Class Sparsity Based Supervised Encoding[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1273-1280.

[7] Wen Y. D, Zhang K. P, Li Z. F, et al. A Discriminative Feature Learning Approach for Deep Face Recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence , 2016, 47(9): 499-515.

[8] Liu W. Y, Wen Y. D, Yu Z. D, et al. Large-Margin Softmax Loss for Convolutional Neural Networks[C]. Proceedings of The 33rd International Conference on Machine Learning, 2016: 507-516.

[9] Rezaei M, Nafchi HZ, Morales S. Global Haar-Like Features: A New Extension of Classic Haar Features for Efficient Face Detection in Noisy Images [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 8333:302-313.

[10] He K. M, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[11] Sandler M , Howard A , Zhu M , et al. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation[J]. Computer Science, 2018, 23(4): 2103-2112.