2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE) | 978-1-6654-2709-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICBASE53849.2021.00091

Handwriting Recognition Based on Resnet-18

Xiaoran Chi
School of Computer Science and Technology
University of Jinan
Jinan, China
1163414243@qq.com

Jiayu Li
School of Mechanical Electronic and Information Engineering
China University of Mining & Technology Beijing
Beijing, China
yu123456788889@163.com

Shuqi Huang*
School of Computer and Science
Tianjin University
Tianjin, China
*Guanghua.ren@gecacademy.cn
†These authors contributed equally.

***Abstract*—With the development of picture recognition, the number of models of recognizing hand-written images are increasing explosively. In order to find higher-precision and better solutions, more and more people begin to pay attention to improving such models. The complex structure and performance loss of many models limits the growth of Handwriting Recognition. To tackle this issue, we employ Resnet-18 whose structure is optimized and simpler. In the experiment, using Resnet-18 to train this model can not only consider the accuracy, but also ensures that the quantity of parameters is acceptable. With the assist of OpenCV, the numbers can be located by a frame straightforward. The proposed model achieved 99.3% Accuracy when it was tested with 10000 images in the MINIST dataset. Furthermore, this mode can recognize 100 numbers (written by human) almost without making a mistake.**

Keywords-Resnet-18; Handwriting recognition; Higher ACC; OpenCV

I. INTRODUCTION

Handwritten digital recognition is a classic and meaningful recognition technology, which has high practical application value. Although a lot of work has been done in handwritten digital recognition, there is still room for higher accuracy and efficiency. The most common networks for digital recognition are convolutional neural networks [1, 2], however, when using the gradient descent algorithm, it is easy to make the training results converge to the local minimum rather than the global minimum. This method has difficulty in describing the character. Also, there are many people using quantum neural network to realize the method of number recognition [3]. This method has difficulty in describing the character feature and the strong noises can make the character feature lose or transfer [4]. For effectively solving the problem of difficulty in training when the number of neural network layers is increasing, Resnet adds a short circuit between the two convolutional layers.

Many works are devoted to handwritten recognition by Convolutional Neural Network (CNN). For instance, utilizing a squeeze-extracted multi-feature convolution neural network (SE-MCNN) to handwritten formula symbols has got a lot of heat [5]. For instance, Wu *et al.* proposed a new method for handwritten word recognition by combining position embeddings with residual networks and bidirectional long short-term memory networks, and the proposed model achieved the best result on two public corpora without additional language resource [6]. Furthermore, Multi-Language Online Handwriting Recognition combines the state-of-the-art components with novel additions in a flexible framework, which allows them to easily transfer improvements between languages and scripts. For this special structure and methods, it can be easily used on powerful machines as well as mobile devices with limited computational power by just changing some settings of the system [7-10].

In this paper, we will exhibit our main contributions which can be summarized as follows:

1.We adapt Resnet-18 in addressing handwriting recognition task.

2.The performance of Resnet-18 model on locating handwritten numbers can be improved by utilizing OpenCV.

3.The proposed model based on Resnet-18 and OpenCV achieved superior accuracy in comparison with other methods.

Based on the existing research at home and abroad, the rest of this paper is organized as follows. In Section 2, the training and testing on Resnet-18 and OpenCV are introduced in detail. In Section 3, the performance between Resnet-18 with OpenCV and other models is compared and analyzed. Finally, Section 4 is the conclusion of this paper.

II. MODEL FORMULATION

A. Formulation of ResNet-18 Model

1).Basic Block

In the Basic Block, the weight layer can be regarded as the convolution layer and is learned by x through these weight layers. The feature of this structure is the shortcut added outside the two layers, which can output the form of.

978-1-6654-2709-8/21/$31.00 ©2021 IEEE
DOI 10.1109/ICBASE53849.2021.00091

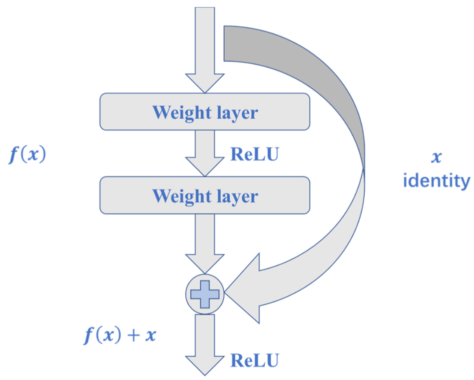

Fig. 1 Basic Block of Resnet-18.

2).The Meaning of Shortcut Block

In the past, it is known that with the increasing number of layers, the model will be harder to train, but with the improvement of this structure, we can solve the vanishing gradient.

$$\frac{dloss}{dxl}=\frac{dloss}{dxL}*\frac{dxL}{dxl}=\frac{dloss}{dxL}(1+\frac{d\sum_{i=1}^{L-1}(xi,wi)}{dxl}) \quad (1)$$

In the whole formula, "1" indicates that shortcut can inherit gradient unconditionally, when the formula $d\sum_{i=1}^{L-1}(xi,wi)/dxl$ is closed to zero, the model can still keep the gradient when the number of network layers is small, so basic block can effectively solve the problem that the model is difficult to train due to the increase of network layers.

3).Whole model

Resnet-18 consists of the build-up layer in the beginning, four Resblocks (each Resblock contains two basic blocks and each basic block contains two build-up layers), and the last fully connected layer, so the number of layer is 1+(4*4)+1=18.

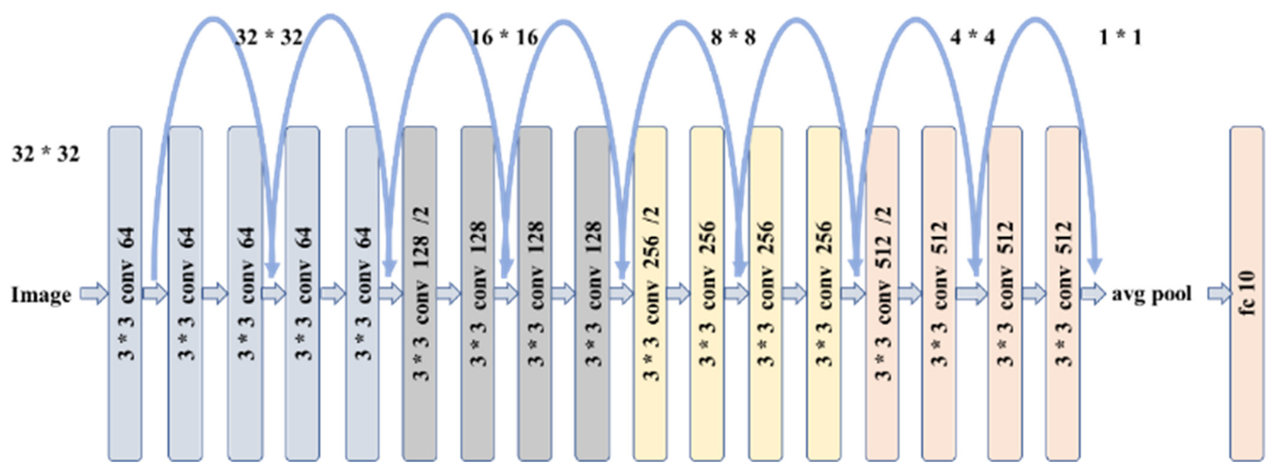

Fig. 2 Main Structure of Resnet-18.

B. Formulation of OpenCV Model

1).Overview of OpenCV

OpenCV is an open source, cross-platform computer vision library that runs on multiple operating systems, including Windows, Linux, and Mac OS. OpenCV is written in C ++ and retains C, Python, Java and other interfaces, it is a lightweight and efficient open source library. As a basic open source project of image processing and computer vision, OpenCV has been widely used in many fields, such as face recognition, motion tracking, object recognition and so on. In this paper, it is mainly applied to handwritten digit recognition.

OpenCV is made up of many modules, which can be divided into many layers:

1.The bottom layer is all kinds of hardware optimization based on the hardware acceleration layer.

2.The next layer is the OpenCV_Contrib module, which contains OpenCV, code contributed by other developers, that contains most of the high-level functional functionality. This is the core of OpenCV.

3.The next layer is the language binding and the sample application.

4.The top layer is the interaction between OpenCV and the operating system.

Tab. 1 The structure of OpenCV

Operating System	Windows, Linux, OSX, Android, iOS
Bindings	Python and Java
Application	Samples, Apps, Solutions
OpenCV Contrib	Face, text, rgbd
OpenCV	Imgproc and objdetect
OpenCV HAL	SSE, NEON, IPP, OpenCL, CUDA, OpenCV4Tegra

2).Graying and Binaryzation of Image

Compared with color image, black and white image processing is much easier, so we need to gray the image. This paper uses the interface of OpenCV visual library. Cv2. CvtColor function and RGB2GRAY parameter are used to transform RGB color image into gray image.

Image binarization is to set the grayscale value of the pixels on the image to 0 or 255, that is, the whole image to present a clear black and white effect. In order to binarize RGB color image, the first step is to grayscale the image. In the digital image processing, binary image plays a very important role. The binaryzation of image greatly reduces the amount of data in the image, which can highlight the contour of the target. In this paper, the function cv. threshold is used to binaryze the image.

3).Image Smoothing

Usually, the noise interference phenomenon will appear in the process of image collection and preservation, which will affect the image processing. Therefore, noise should be eliminated before image analysis. OpenCV noise elimination methods include: mean filtering, median filtering and Gaussian filtering. In this paper, Gaussian filter is used, which is a kind of linear smoothing filter and has a good effect on removing Gaussian noise.

Gaussian filtering is the process of weighted average of the whole image, and the value of each pixel is obtained after weighted average of its own and other pixel values in the neighborhood. The specific operation of Gaussian filtering is to scan each pixel in the image with a template, and replace the value of the pixel in the center of the template with the weighted average gray value of the pixel in the neighborhood determined by the template.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{2}$$

where x is the distance from the current value, σ is the standard deviation, and $G(x)$ is the weight value.

The smoothing effect of Gaussian filter is soft, and the edge information in the image can be retained. Considering that this paper is to identify the numbers, the sharp information needs to be retained in the process of image processing, so Gaussian filtering is adopted.

4).Drawing Contours

The contour can be considered as a curve with continuous points of the same color or gray. Contour is very useful in shape analysis as well as object detection and recognition. In this paper, the function cv2.findContours is used to draw the contour of the image. The function cv2.findContours has three parameters, the first is the input image, the second is the contour retrieval mode, and the third is the contour approximation method.

We set the third parameter to cv2.chain_approx_simple. It removes all redundant points from the outline, compressing the outline and saving memory. Use the rectangle in the image below to demonstrate this technique. Draw a blue circle at each coordinate in the outline list. The first figure shows the effect with cv2.CHAIN_APPROX_NONE, and the second figure shows the effect with cv2.CHAIN_APPROX_SIMPLE, with only four points.

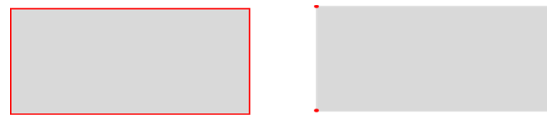

Fig. 3 Compare the results of cv2.CHAIN_APPROX_NONE and cv2.CHAIN_APPROX_SIMPLE

5).Advantages

As an open-source software, OpenCV has more than 400 free image processing functions, involving a wide range, including image processing, pattern recognition, motion video, and three-dimensional reconstruction, covering most of the application fields of machine vision. Many of its algorithms also do a lot of optimization, such as threshold segmentation, edge extraction.

III. Experiments

A. Datasets

In the experiment, we decided not to select the same data set in the process of training and testing to reflect the robustness of the model. We selected 10000 handwritten digital pictures from MINIST dataset to train the model, while, as for the testing dataset, we wrote 100 numbers in different font and photograph them. Due to the difference between the two datasets, we need to preprocess the test images to a size of 28x28, and utilize OpenCV to change RGB images to grayscale images. The MINST dataset is listed in Tab. 2.

Tab. 2 The dataset utilized in experiment

Dataset	Numbers	Clusters	Dimensions
MINIST	70000	10	784

B. Evaluation Metrics

The experimental results demonstrated that the proposed mode achieved 99.3% ACC on the 10000 test images in MINIST dataset. Furthermore, this model can classify 100 handwritten digital samples perfectly, and ACC reached 99%.

$$ACC = \frac{N_{Right}}{N_{All}} \tag{3}$$

where N_{Right} indicates the number of samples whose label is the same as predicted result, and N_{All} indicates the number of all the samples.

C. Experimental Results and Analysis

The training of the model will adopt 2, 5, 10 and 15 epochs each time and compare the changes of ACC. Then according to the variance between the last two trains, we can find the training method that consumes the shortest time and ensures the precise rate.

Tab. 3 The first group of training loss and testing *ACC* under four different epochs

	2 epochs	5 epochs	10 epochs	15 epochs
Training Loss	0.0839	0.0055	0.0012	0.0003
Testing *ACC*	0.9905	0.9929	0.9927	0.9935

Tab. 4 The second group of training loss and testing *ACC* under four different epochs

	2 epochs	5 epochs	10 epochs	15 epochs
Training Loss	0.0180	0.0053	0.0013	0.0002
Testing *ACC*	0.9905	0.9929	0.9940	0.9942

As we can see from the two Tabs above, when the number of epochs varies from 2 to 5, the accuracy rate increases considerably. By contrast, the variation of testing ACC is gentler when the epochs are 10. Nevertheless, the gap of experiment results between 10 and 15 epochs has almost disappeared. According to the results, 10 epochs will be chosen to train the model based on Resnet-18 for its shorter time consumed and higher ACC.

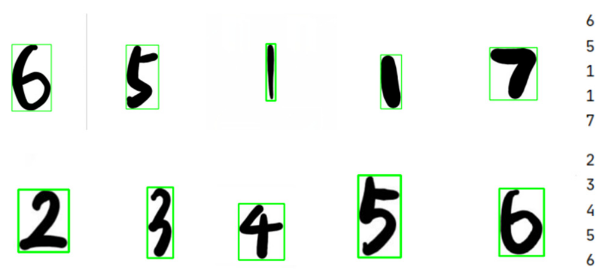

Fig. 4 Samples output and the correct predicted values of samples on the right.

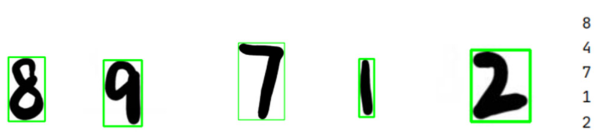

Fig. 5 Error identification number “9” as number “4”.

The accuracy of model recognition after training is very high. The recognition of 100 handwritten digits has been tested 3 times and only one error has occurred. The model identifies a “9” as a “4”, and indeed the handwritten 9 bears a slight resemblance to 4 in shape. We can still consider it a model with higher accuracy in recognize handwritten numbers.

IV. Conclusion

This paper proposed a model based on ResNet-18 and OpenCV to recognize handwritten numbers. To certify the accuracy of the method, a test data set consisting of 100 handwritten numbers is created, and the performance of the model is proved nearly perfect. In contrast, the sample methods utilizing common CNN are not likely to acquire such a higher accuracy up to 99.3%. Moreover, the use of OpenCV improves the capability of our model through locating handwritten numbers precisely.

In the future, we will focus on reducing the time of calculation required for model training. Distributed computing frameworks may be the breakthrough to solve this problem, like Spark. Furthermore, the ResNet-18 and OpenCV models will be upgraded to recognize both letters and numbers. For instance, the model will get involved to recognize verified codes on websites.

References

[1] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," Proc. of the European Conference on Computer Vision, pp. 818-833, 2014.

[2] J. Schmidhuber, "Deep learning in neural networks: an overview," Neural Networks, vol. 61, pp. 85-117, 2015.

[3] F. Li, S. Zhao, and B. Zheng, “Quantum neural network in speech recognition,” 6th International Conference on Signal Processing, Vol. 2. pp. 120-128, 2002.

[4] D. Zhu and R. Wu, “A multi-layer quantum neural networks recognition system for handwritten digital recognition,” Third International Conference on Natural Computation (ICNC 2007), vol. 1, pp. 718-722, 2007.

[5] D. Fang and C. Zhang, "Multi-Feature Learning by Joint Training for Handwritten Formula Symbol Recognition," IEEE Access, vol. 8, pp. 48101-48109, 2020.

[6] X. Wu, Q. Chen, J. You and Y. Xiao, "Unconstrained Offline Handwritten Word Recognition by Position Embedding Integrated ResNets Model," IEEE Signal Processing Letters, vol. 26, no. 4, pp. 597-601, 2019.

[7] D. Keysers, T. Deselaers, H. A. Rowley, L. Wang and V. Carbune, "Multi-Language Online Handwriting Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1180-1194, 2017.

[8] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, “A Novel Connectionist System for Unconstrained Handwriting Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 855-868, 2009.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 770-778, 2016.

[10] M. Arun, S. Arivazhagan, and D. Rathina, “Handwritten Text Segmentation Using Pixel Based Approach,” 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019.