Offline Handwritten Chinese Character Recognition Based on an Improved Inception: Postprint

Authors: Chen Zhan, Qiu Weigen, Zhang Lichen

Date: 2019-01-28T00:00:00+00:00

Abstract

Due to the complexity and variability of character shapes, offline handwritten Chinese character recognition has long been a challenging problem in pattern recognition. The development of deep convolutional neural networks has provided a direct and effective solution. This study investigates offline handwritten Chinese character recognition based on inception-structured neural networks and proposes an improved inception architecture featuring a simpler structure, easier network depth expansion, and fewer training parameters. The method was experimentally validated on the CISIA-HWDB1.1 dataset using the stochastic gradient descent optimization algorithm, and the model achieved an average accuracy of 96.95%. Experimental results demonstrate that the improved inception architecture exhibits better robustness in image classification and is more easily extensible to other application domains.

Full Text

Preamble

Vol. 37 No. 4
Application Research of Computers

Offline Handwritten Chinese Character Recognition Based on an Improved Inception

Chen Zhan, Qiu Weigen, Zhang Lichen
(School of Computers, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Due to the complexity and variety of glyphs, offline handwritten Chinese character recognition has long been a challenging problem in pattern recognition. The development of deep convolutional neural networks provides a direct and effective solution to this problem. This paper investigates offline

handwritten Chinese character recognition based on Inception neural networks and proposes an improved Inception structure that offers a simpler architecture, easier network depth expansion, and fewer training parameters. The method was evaluated on the CISIA-HWDB1.1 dataset using stochastic gradient descent optimization, achieving an average accuracy of 96.95%. Experimental results demonstrate that the improved Inception structure exhibits better robustness in image classification and can be easily extended to other application domains.

Keywords: offline handwritten Chinese characters; convolutional neural network; inception

Classification: TP391.43
DOI: 10.19734/j.issn.1001-3695.2018.09.0784

0 Introduction

Since the 1980s, handwritten Chinese character recognition (HCCR) has been an important yet challenging research area in pattern recognition [1]. The primary difficulties in handwritten Chinese character recognition stem from the large number of character categories, complex font structures, significant glyph variations, diverse writing styles, and particularly the existence of numerous highly similar characters with extremely subtle differences, such as "已-己", "口-口", "沮-泪-汩", which pose great challenges for automated computer recognition [2].

Through years of dedicated research efforts, HCCR has made substantial progress. Reference [3] employed discriminative feature extraction (DFE) and discriminative learning quadratic discriminant function (DLQDF) classifiers, achieving the best recognition rates of 94.20% (DB1.0), 92.08% (DB1.1), and 92.72% (ICDAR 2013 Competition DB) on several subsets of the CASIA-HWDB offline handwritten Chinese character dataset.

In recent years, deep learning has gained widespread attention from both academia and industry, achieving remarkable success in computer vision and image recognition, and bringing new vitality and highly effective solutions to the HCCR problem. Typical deep learning architectures include Deep Belief Networks (DBN), Stacked Auto-Encoders (SAE), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). In recent years, research on deep convolutional neural networks has achieved a series of breakthroughs in image classification [4-9], making them an important tool for solving offline handwritten Chinese character recognition. References [10, 11] investigated CNN-based HCCR methods and achieved promising results on both CASIA-HWDB1.0 and CASIA-HWDB1.1. However, CNN network structures are complex, and optimizing fully connected layers requires massive training data and computational resources. Higher accuracy demands deeper CNN networks, which simultaneously require more complex architectures to

counteract the inevitable negative effects of gradient vanishing and gradient explosion.

References [8, 12–14] proposed the Inception structure and applied it to HCCR. The Inception structure offers smaller parameter counts and better robustness. However, Inception remains structurally complex, making it difficult to stack to great network depths [9]. This paper proposes a CNN network based on an improved Inception structure, temporarily referred to as Joint-Net for convenience. Joint-Net not only inherits the advantages of Inception—good generalization performance and small parameter count—but also enables internal network deepening and performance improvement without causing gradient vanishing or gradient explosion. Our experiments demonstrate that it achieves high average accuracy and can be easily extended to other application domains.

1 Related Work

Inception was first proposed in GoogleNet [8], where the 22-layer GoogleNet won the 2014 ImageNet competition, achieving a Top-5 error rate of 6.67% on the ImageNet dataset. Reference [13] introduced the second version of Inception, incorporating Batch Normalization (BN) layers to reduce the model' s error rate on ImageNet to 4.9%. Reference [12] proposed the third Inception version. In Inception v3, 3×3 convolutions were decomposed into 1×3 and 3×1 kernels, reducing network parameters while improving recognition performance, lowering the Top-5 error rate on ImageNet to 3.5%. Reference [14] combined ResNet [9] architecture with enriched Inception structures, achieving a 3.08% Top-5 error rate on ImageNet. The proposal and improvement of Inception have significantly enhanced the recognition performance of deep CNN networks.

The success of the Inception structure benefits from extensive use of 1×1 convolution kernels and multi-level feature transmission. A typical Inception structure is shown in Figure 1 [Figure 1: see original paper]. In Figure 1, the 1×1 convolution kernels can increase network depth and transform feature dimensions (for dimensionality reduction or increase) with only minimal additions to computational cost and parameter count. Additionally, the Inception structure contains subnetworks with stacked convolutional layers of depths 5, 3, 2, and 1. The depth-5 subnetwork significantly increases network depth, while the depth-1 subnetwork allows features to reach the next Inception structure more quickly, mitigating gradient vanishing and explosion caused by increased network depth. Subnetworks of different depths provide features at different levels, enhancing the network' s scale generalization performance.

However, the complex Inception structure hinders the stacking of very deep networks, while deeper networks often exhibit better performance when gradient vanishing and explosion do not occur [9, 15–17]. This paper simplifies and improves the Inception structure while retaining its advantages, making network depth expansion easier and improving network performance. Based on this improved Inception structure, we propose the Joint-Net network. Joint-Net

extensively stacks the improved Inception structures to increase network depth and removes the final fully connected layer, thereby improving recognition performance while preventing a substantial increase in parameter count with the number of categories.

2 Joint-Net Network Architecture

2.1 Improved Inception Structure

As shown in Figure 2 [Figure 2: see original paper], the Unit structure in (a) consists of convolutional layers with 1×1 and 3×3 kernels, with the outputs of the two convolutions concatenated along the channel dimension as the Unit' s output. The Unit structure replaces the 3×3 convolutional layer to retain the advantage of Inception' s small parameter count. Figure 2(b) illustrates the improved Inception structure. Let the number of Units be N. On one hand, structure (b) contains subnetworks with depths $N, N-1, \ldots, 1$, preserving Inception' s scale adaptability. On the other hand, for each Unit, there is direct input from the Base and direct output to the next layer, which effectively avoids gradient vanishing and explosion phenomena. Consequently, the depth of structure (b) can theoretically be very deep. Finally, all Unit outputs in (b) are concatenated along the channel dimension as input to the 1×1 convolution, leveraging the advantages of 1×1 kernels like the original Inception structure.

In Figure 2(b), except for the first Unit whose input comes from the Base, each Unit' s input is the concatenation of the Base and the previous Unit' s output along the channel dimension. Except for the last Unit whose output only propagates directly to the 1×1 convolution layer, each Unit' s output is copied and passed to the next Unit. The improved Inception structure retains Inception's advantages, enhances the network's scale adaptability, and facilitates convenient stacking of network depth.

2.2 Fully Convolutional Classification Module

Deep convolutional neural networks employ one-hot encoding for category representation at the output layer. Let the number of categories be n and the category label be i, then the corresponding one-hot vector A is defined as $\langle MATH_0 \rangle$. Since the output layer uses one-hot encoding, the number of artificial neurons in the output layer equals the number of categories, leading to a space complexity of $\langle MATH_1 \rangle$ for fully connected layer parameters. For example, considering 3,755 frequently used Chinese characters, with both input and output neurons numbering 3,755 and each parameter occupying 4 bytes, the output fully connected layer alone occupies over 107 MB of space. In practical applications, networks may require multiple fully connected layers to improve recognition performance, with each layer consuming even more parameters.

This paper replaces fully connected layers with convolutional layers for classification. Representing feature map shape as (number of feature maps, height,

width), let the shape of the last convolutional layer's feature maps be $\langle MATH_2 \rangle$ for n categories. A fully connected classification layer would require $\langle MATH_3 \rangle$ parameters. Using a convolutional layer as the classification layer requires the output feature map shape to be $\langle MATH_4 \rangle$, where we adjust $\langle MATH_5 \rangle$ to satisfy $\langle MATH_6 \rangle$. Thus, a convolutional classification layer requires only $\langle MATH_7 \rangle$ parameters. Consequently, a single fully connected classification layer has $\langle MATH_8 \rangle$ times more parameters than a convolutional classification layer.

Replacing fully connected layers with a single convolutional layer may degrade network recognition performance. Joint-Net addresses this by combining multiple convolutional layers to adjust the network, enabling convolutional output layers to achieve classification performance comparable to fully connected layers.

2.3 Joint-Net Construction

The improved Inception structure facilitates easy network construction. The network' s main body can be completed by simply stacking multiple improved Inception structures. As shown in Figure 3 [Figure 3: see original paper], the 1×1 convolutional layers serve a connecting role in the network architecture, which we call "joints." Pooling within Inception is inconvenient, so optional pooling layers are incorporated into the joints, performing pooling operations when needed.

To achieve better network performance, this paper adds BN and ReLU layers to both Units and joints, and includes max pooling layers in joint modules. Since exclusively using 1×1 and 3×3 kernel combinations to replace 3×3 kernels does not perform well, but achieves excellent results after adding 3×3 kernels, we include 3×3 as an additional option for flexible network adjustment and improved performance. Detailed designs of Units and joints are shown in Figure 4 [Figure 4: see original paper].

During training, data augmentation is applied to samples, including padding with 4 zero pixels on boundaries, random cropping to 32×32, random horizontal flipping, and normalization.

Table 1 Network Structure

Unit Count	Kernel Size	Input Channels/Unit	Output Channels/Unit	Pooling Layer
Joint0	1 or 3	-	-	MaxPool
Units1	-	-	-	-
Joint1	1 or 3	-	-	MaxPool
Units2	-	-	-	-
Joint2	1 or 3	-	-	MaxPool
Units3	-	-	-	-
Joint3	1 or 3	-	-	MaxPool

Unit Count	Kernel Size	Input Channels/Unit	Output Channels/Unit	Pooling Layer
Units4	-	-	-	-
Joint4	1 or 3	-	-	MaxPool
Units5	-	-	-	-
Joint5	1 or 3	-	-	MaxPool
Units6	-	-	-	-
Joint6	1 or 3	-	-	MaxPool
Units7	-	-	-	-
Joint7	1 or 3	-	-	MaxPool

Figure 5 [Figure 5: see original paper] illustrates the training process of Joint-Net on the CASIA-HWDB1.1 training and test sets. Joint-Net is trained on the CASIA-HWDB1.1 training set and evaluated on the test set. The results show clear accuracy improvements at each learning rate decay, with the network eventually converging. A comparison of experimental results with other network models is presented in Table 2 , where entries marked with "*" indicate results reproduced by us as the original literature did not report separate training on CASIA-HWDB1.1.

Table 2 Comparison of Joint-Net Accuracy with Other Models on CASIA-HWDB1.1

Network	Params	CASIA-HWDB1.1 (Accuracy %)
DirectMap+ConvNet+Adaptation[10] *	23.5M	-
HCCR-CNN12Layer+GSLRE 4X[11] *	32.7M	-
Joint-Net	-	96.95

3 Experiments

To validate the effectiveness of the Joint-Net model for offline handwritten Chinese character recognition, this paper selects the large-scale CASIA-HWDB1.1 dataset. The offline handwritten Chinese character dataset CASIA-HWDB1.1 includes 3,755 GB2312-80 level-1 frequently used Chinese characters. The training set contains handwriting from 240 writers, while the test set contains handwriting from 60 writers, totaling 1,121,749 samples, making it a large-scale pattern recognition dataset. All images in the dataset are resized to 32×32 for recognition experiments.

All experiments in this paper are implemented and executed using PyTorch 0.4 on a Windows 10 64-bit system. The hardware environment consists of an Intel i7-6700K 4.0GHz CPU, 8GB DDR4 RAM, and an NVIDIA GTX 1080 8GB GPU.

To verify Joint-Net' s performance, extensive repeated experiments are conducted on CASIA-HWDB1.1. The experiments employ an identical network architecture comprising 7 stacked improved Inception structures, 8 joints, and 1 convolutional layer. The network structure is detailed in Table 1, where Units represent the Unit components within corresponding Inception* modules, and Conv* layers denote convolutional layers with dropout used for classification.

The training strategy adopts the following settings:
a) Training epochs: 60
b) Batch size: 128
c) Learning rate schedule: Initial learning rate $lr = 0.1$, decayed to 0.02 at epoch 20, to 0.004 at epoch 40, and to 0.0008 at epoch 50
d) Weight decay: 0.0005
e) Optimizer: Nesterov-accelerated SGD with momentum $\langle MATH_9 \rangle$

Despite having deeper convolutional layers and a large number of categories (3,755), Joint-Net maintains a small parameter count by removing fully connected layers, and this parameter count does not increase rapidly with more categories. The model demonstrates better robustness and generalization capability, showing significant improvement over results reported in [10, 11] on CASIA-HWDB1.1.

To verify Joint-Net' s practicality, we apply it to a Chinese subtitle extraction system, achieving high recognition rates that demonstrate its suitability for real-world applications. The Chinese subtitle extraction model is shown in Figure 6 [Figure 6: see original paper].

In Figure 6, a pre-trained Joint-Net model processes images containing Chinese subtitles to produce scores for 3,756 categories (including 1 background category and 3,755 frequently used Chinese character categories). Since subtitle position is not of interest and only subtitle information is extracted, no coordinate regression is performed. To illustrate the Infer process, let Class scores be tensor X and Class map be tensor Y, where maxpool is a 3×3 max pooling operation with stride 1, and argmax finds the channel containing the maximum value at each pixel. The Infer process is defined as $\langle MATH_1 0 \rangle$.

Because the Infer operation in Figure 6 takes local maxima of class scores—equivalent to multiple voting—the accuracy is higher than single-character recognition. However, classifying each 3×3 region centered at every pixel reduces speed. Using the model in Figure 6, testing on 200 actual video frames containing Chinese characters achieves a recall rate of 98.9%, accuracy of 98.4%, and processing speed of 14 frames per second.

Experimental results demonstrate that Joint-Net achieves state-of-the-art single-model performance on the 3,755-category offline handwritten Chinese character dataset CASIA-HWDB1.1. This indicates that Joint-Net is a powerful convolutional neural network architecture with excellent robustness and generalization capabilities. Its unique Units and joints effectively increase network depth while enhancing robustness and generalization, enabling the network to more easily

achieve superior results. The successful application in a practical Chinese subtitle extraction system demonstrates Joint-Net' s practical value. Due to its structural simplicity, network model compression algorithms [1][8] can be readily applied to Joint-Net, representing a promising direction for future research.

References

[1] Zhao Jiyin, Zheng Ruirui, Wu Baochun, et al. A review of offline handwritten Chinese character recognition [J]. *Acta Electronica Sinica*, 2010, 38(2): 405-415.

[2] Jin Lianwen, Zhong Zhuoyao, Yang Zhao, et al. Applications of deep learning for handwritten Chinese character recognition: a review [J]. *Acta Automatica Sinica*, 2016, 42(8): 1125-1141.

[3] Liu Chenglin, Yin Fei, Wang Dahan, et al. Online and offline handwritten Chinese character recognition: benchmarking on new databases [J]. *Pattern Recognition*, 2013, 46(1): 155-162.

[4] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition [J]. *Neural Computation*, 1989, 1(4): 541-551.

[5] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. *Proceedings of the IEEE*, 1998, 86(11): 2278-2324.

[6] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [EB/OL]. (2015-04-10). https://arxiv.org/abs/1409.1556.

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

[8] Szegedy C, Liu Wei, Jia Yangqing, et al. Going deeper with convolutions [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2015: 1-9.

[9] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition [C]//Proc of the IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2016: 770-778.

[10] Zhang Xuyao, Bengio Y, Liu Chenglin. Online and offline handwritten Chinese character recognition: a comprehensive study and new benchmark [J]. *Pattern Recognition*, 2017, 61: 348-360.

[11] Xiao Xuefeng, Jin Lianwen, Yang Yafeng, et al. Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition [J]. *Pattern Recognition*, 2017, 72: 72-81.

[12] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception architecture for computer vision [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2016: 2818-2826.

[13] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C]//Proc of International Conference on Machine Learning. 2015: 448-456.

[14] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, Inception-ResNet and the impact of residual connections on learning [EB/OL]. (2016-08-23). https://arxiv.org/abs/1602.07261.

[15] Srivastava R K, Greff K, Schmidhuber J. Highway networks [J]. arXiv preprint arXiv:1505.00387, 2015.

[16] Srivastava R K, Greff K, Schmidhuber J. Training very deep networks [C]//Advances in Neural Information Processing Systems. 2015: 2377-2385.

[17] Huang Gao, Liu Zhuang, Van Der Maaten L, et al. Densely connected convolutional networks [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2017.

[18] Hu Jie, Shen Li, Albanie S, et al. Squeeze-and-excitation networks [EB/OL]. (2018-10-25). https://arxiv.org/pdf/1709.01507.pdf.

Note: Figure translations are in progress. See original paper for figures.