Convolution Neural Networks for Chinese Handwriting Recognition

Xu Chen
Stanford University
450 Serra Mall, Stanford, CA 94305
xchen91@stanford.edu

Abstract

Convolutional neural networks have been proven powerful in handwritten digits and alphabetic recognition. In this project, different convolutional neural networks are explored to classify handwritten Chinese characters. This paper compares the performance of different architecture on the CASIA offline Chinese handwriting database. It also analyzes the effect of gradient feature extraction with the same set of architectures. Experimental results show that deeper networks with larger number of filters give better accuracy, and gradient feature results in better performance in most networks compared to the raw image data.

1. Introduction

Various forms of classification techniques have been extensively researched to solve the image recognition problem, a subfield of which is offline handwriting recognition. Researches on recognizing digits and alphabets have taken several paths. One is to design deep convolutional neural networks (CNN) on the preprocessed image data. Another is to train linear or quadratic classifiers with extracted features from the images. Both techniques have been proven powerful in terms of accuracy. However, digits and alphabets are still relatively simple compared to handwritten ideographic characters, such as Chinese. This project examines how well CNNs perform in classifying handwritten Chinese characters, and explores the improvement of feature extraction in combination with CNNs.

Recognizing Chinese characters is a more challenging task than recognizing digits and alphabets. Chinese characters have way more classes, with more than 7000 characters in the common vocabulary, compared to 10 in digits and 52 in English alphabets. Chinese characters also have more strokes, which form more complex structures. For example, 啊 consists of 10 strokes and shows left to middle and right structure, while 藹 has 14 strokes and displays above to below structure. Furthermore, Chinese handwriting shows much more variance and deformation due to different writing styles. Figure 1 shows the complex structures and the extent of deformation in Chinese handwriting.

This paper first explores different CNN architectures by varying the depth and layer structures and comparing the performance. We analyzed the impact of increasing the number of layers or filters, as well as adding a convolutional layer versus a fully connected layer. The second part applies feature extraction techniques to CNNs. It is well acknowledged that the directions of character strokes contain important information for character recognition. This project exploits gradient feature extraction technique proposed by Liu [1] and analyze the improvement by feeding feature matrices as inputs to CNNs instead of preprocessed character images.

The rest of the paper is organized as follows. **HERE**

2. Related Work

Quite some researches have focused on recognizing ideographic handwritings such as Chinese, most of which propose advanced techniques on data preprocessing. J. Tsukumo et al [2] and Yamada et al [3] proposed two similar techniques to mitigate the deformation due to writing styles by equalizing the line density. Because the former is more efficient while giving similar results as the latter, it is employed as a part of data preprocessing in this paper. The other important technique is extracting the direction of strokes at each pixel. One popular idea is the directional elemental feature (DEF), also called direction histogram code [4] due to its promising performance. However, Liu [1] proposed a more efficient gradient extraction technique, which avoids computing gradient magnitude and angle, which is also applied as a part of the data, preprocess. However, all of the researches mentioned above fall back to linear or quadratic classification after the advanced data preprocessing or feature extraction techniques.

Figure 1. Handwriting samples from 3 writers

	#Writers	#Classes	#Chinese Samples	#Symbol Samples	#Total Samples
HWDB1.0	420	3,866	1,609,136	71,122	1,680,258

Table 1. Summary of dataset

Figure 2. Image after each step of preprocess

3. Data

3.1. Dataset

This project uses CASIA offline Chinese handwriting database, namely CASIA-HWDB1.0. It consists of 3,866 Chinese characters written by 420 different writers as shown in Table 1. Each data file is a collection of samples with 10 byte headers followed by the gray-scale matrices. The full dataset consists of a large training set and a smaller test set. The test set contains approximately 80 randomly selected images from each category and the training set contains the rest. A part of the training set is also reserved as the validation set, which contains 40 randomly selected images from each category and leaves 300 images per category for the real training set.

Due to the limitation of computation resources, most of the experiments are conducted on a subset of the full dataset. The subset consists of all samples of 300 randomly chosen categories, which results in a training set of 90,000 samples, a validation set with 12,000 samples and a test set with 24,000 samples.

3.2. Preprocessing

Raw data is first parsed and converted to .bmp gray-scale images and then processed in four steps: standardization, resizing, line-density equalization and smoothing. 1) The gray-scale image is first inverted and then standardized by mapping the minimum and maximum value to 0 and 255 respectively. 2) Then, the image is resized with either contour-center or gravity-center size normalization. The true boundary of the strokes is determined and the redundant background area is cropped. Then the image is resized to a standard size with center of gravity or contour as the center of the resized image. Using center of gravity is more resilient to noise but harder to retain the entire image, so center of contour is used for most of the samples. After visual inspection of the characters, we determined 64x64 to be a proper size. 3) Because handwritten Chinese characters

A	B	C	D	E	F	G	H	I
7 weight layers	7 weight layers	7 weight layers	9 weight layers	9 weight layers	9 weight layers	11 weight layers	11 weight layers	11 weight layers
Conv3-32	Conv3-64	Conv3-64	Conv3-32	Conv3-64	Conv3-64	Conv3-32 Conv3-32	Conv3-64 Conv3-64	Conv3-64 Conv3-64
Pool2								
Conv3-64	Conv3-128	Conv3-128	Conv3-64 Conv3-64	Conv3-128 Conv3-128	Conv3-128 Conv3-128	Conv3-64 Conv3-64	Conv3-128 Conv3-128	Conv3-128 Conv3-128
Pool2								
Conv3-128	Conv3-256	Conv3-256	Conv3-128 Conv3-128	Conv3-256 Conv3-256	Conv3-256 Conv3-256	Conv3-128 Conv3-128	Conv3-256 Conv3-256	Conv3-256 Conv3-256
Pool2								
Conv3-256 Conv3-256	Conv3-512 Conv3-512	Conv3-512	Conv3-256 Conv3-256	Conv3-512 Conv3-512	Conv3-512	Conv3-256 Conv3-256 Conv3-256	Conv3-512 Conv3-512 Conv3-512	Conv3-512 Conv3-512
Pool2								
		FC-1024			FC-1024			FC-1024
FC-1024								
FC-300								
softmax								

Table 2. CNN Configurations

usually contain various deformation in the stroke structures, the line-density nonlinear normalization proposed by Tsukumo et al [2] is further applied. The goal of this normalization is to reduce the shape variation by resampling the pixels based on line density on horizontal and vertical axes. 4) Lastly, the normalized image is smoothed by a 3x3 mean filter. Figure 2 visually illustrates the each step. After all steps of preprocessing, images from different users demonstrate much less variation in magnitude, size, structure and position.

3.3. Feature extraction

To extract gradient feature, 3x3 sobel operators are first applied to the preprocessed image to get the horizontal and vertical grayscale gradient at each pixel. As proposed by Liu [1], L directions with equal intervals are defined and each gradient vector is decomposed into two nearest directions in parallelogram manner. An L-dimensional gradient code is obtained for each pixel. Then the 64x64 normalized image is equally divided into 16x16 subblocks, and the L-dimensional gradient codes are summed within each subblock. Then the resolution of subblocks is downsampled with a Gaussian filter to 8x8, resulting in an 8x8xL gradient feature. Finally a variable transformation y=x^0.4 is applied to make the distribution more Gaussian-like. In this project, L = 8 and 12 are explored in the second part of the experiment. Figure 3 demonstrates the gradient feature extraction and decomposition process.

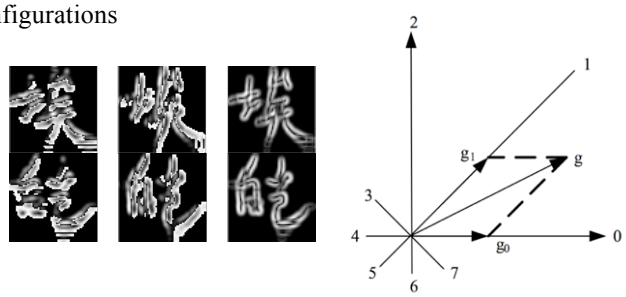

Figure 3. Gradient of 埃 and 皑, and gradient vector decomposition to 8 directions

4. Approach

4.1. Architecture

During training with preprocessed images, the input to CNN is a fixed-size 64x64 grayscale image. The image is passed through a stack of convolutional layers, each of which has 3x3 filters. The convolution stride is fixed to 1 pixel, and the padding is 1 pixel. Spatial pooling is carried out by four max-pooling layers, which follow some of the convolutional layers. Max-pooling is performed over a 2x2 pixel window.

During training with gradient features, the input to CNN is either 8x8x8 or 8x8x12 based on the granularity of gradient decomposition. The overall network structure is maintained but the dimension of the first convolutional layer filter is adjusted for the additional dimension of the input.

	Validation Accuracy			Test Accuracy		
Model	Preprocessed Image	Gradient Feature (8)	Gradient Feature (12)	Preprocessed Image	Gradient Feature (8)	Gradient Feature (12)
A	92.4	92.9	93.2	92.1	92.8	93.1
B	93.5	93.9	94.3	93.1	93.8	94
C	93.1	93.3	94	92.6	93.1	93.4
D	94.6	95	95.3	94.3	94.7	94.1
E	95.7	95.9	96.1	95.5	95.9	95.8
F	95.4	95.6	95.8	95	95.2	95.3
G	95.3	95.3	95.5	94.9	95.1	95.2
H	96.2	96.3	96.4	95.9	96	96,1
I	95.9	95.8	96.2	95.4	95.7	95.8

Table 3. CNN performance on 300-class dataset

4.2. Configurations

To analyze the performance impact of different CNN architectures, the question is dissected into three factors: the depth of the networks, the number of filters in the convolutional layers, as well as the organization of layers.

The design and exploration is conducted largely based on the approach presented by K. Simonyan et al [5]. 9 convolutional neural networks were implemented, and their configurations are outlined in Table 1, one per column. A 7-layer CNN was first implemented as the starting point of exploration. To analyze the performance impact of network depth, 2 layers were added to the base line design in each step, which eventually pushed the depth to 11. To analyze the influence of the number of filters, a variation of the design for each step was added to have double the number of filters in each layer. Lastly, to evaluate the contribution of a convolutional layer versus a fully connected layer, another variation of the design replaced the last convolutional layer with a fully connected layer.

5. Experiment

5.1. Training

Training is conducted on the 300-class subset of the training data. The input is either the preprocessed images, or the extracted gradient features with 8 or 12 directions. The training is carried out by optimizing the multinomial logistic regression function using mini-batch gradient descent with momentum. The batch size was set to 200 and momentum to 0.9. The dropout rate for the fully-connected layers is set to 0.5. The learning rate was initialized to 0.01 and decayed by 10% every epoch.

Validation accuracy is computed on the validation set.

5.2. Testing

Testing is carried out on the provided test set.

5.3. Results and Discussion

The classification results for the 300-class subset are presented in Table 3. The model with best test performance was the 11-layer network with large number of filters and with fine-grained gradient features as input. By comparing the model with the same filter size but different depth (ADG, BEH), we could find that the accuracy improved as the depth of the network increased. Furthermore, the results also show improved performance with larger number of filters and everything else fixed by comparing models AB, DE and GH. Comparing models BC, EF and HI shows that replacing a fully-connected layer with a convolutional layer can improve the performance. Furthermore, by training the model with gradient feature vectors, better results were achieved with the same network architecture, and a finer grain (12 vs 8) resulted in higher accuracy. However, gradient feature vectors seem to have diminishing return based on the depth of the network. It could only provide minimal improvement to a deep network, which explains why it is a technique commonly used for linear or quadratic classifiers.

6. Conclusion

In this project, three factors in CNN architecture were analyzed in terms of their performance impact: number of layers, number of filters and network organization. The influence of gradient feature extraction was also examined with different network configurations. This paper conducted experiments with 9 different convolutional networks on a 300-class subset, either with preprocessed images or extracted gradient features as inputs. Based on the statistics gathered from the test set, we conclude that for convolutional neural networks with small filter sizes: 1) increasing the depth of the network can increase accuracy; 2) increasing the number of filters can increase accuracy; 3) adding a convolutional layer can give better accuracy than adding a fully-connected layer; 3) gradient features can improve performance for relatively shallow networks, but the improvement is minimal when the network is deep enough.

References

[1] Liu, Hailong, and Xiaoqing Ding. "Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes." Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on. IEEE, 2005.A.

[2] J. Tsukumo, H. Tanaka, “Classification of handprinted Chinese characters using non-linear normalization and correlation methods,” Proc. Ninth International Conference on Pattern Recognition, Roma, Italy, 1988, pp. 168–171

[3] H. Yamada, K. Yamamoto, T. Saito, "A nonlinear normalization method for hanprinted Kanji character recognition line density equalization," Pattern Recognition 23(9) (1990) 1023-1029

[4] F. Kimura, T.Wakabayashi, S.Tsuruoka, and Y.Miyake, “Improvement of handwritten Japanese character recognition using weighted direction code histogram,” Pattern Recognition, Vol. 30, No.8, pp. 1329-1337, 1997.

[5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.