Optical Character Recognition using Simple Convolutional Neural Network

Tan D. Lam
HCMC University of Science
1512489@student.hcmus.edu.vn

Huy V. N. Huynh
HCMC University of Science
1512196@student.hcmus.edu.vn

Trieu H. Nguyen
HCMC University of Science
1512595@student.hcmus.edu.vn

Tiep V. Nguyen
HCMC University of Science
nvtiep@fit.hcmus.edu.vn

Triet M. Tran
HCMC University of Science
tmtriet@fit.hcmus.edu.vn

Abstract

In the first decade of 21st century, convolutional neural network is the trend of classification and recognition problem. In this paper, we use a CNN model to recognize printed English characters. This problem can be applied to convert books to editable formats, or any other scanned documents. The program is capable of handling non-ideal images such as noisy, rotated, non-center gravity ones. Recognition accuracy is 100% with ideal images, but ranges between 80 100% with non-ideal images.

1. Introduction

Character recognition is an area with many different applications, such as converting paper documents into text files, or increasing interaction of human and computers. The first method for character recognition is invented in the 1950s, which is template matching. The main idea of this method is comparing character images with a library of prototype images for each character of each font. Template matching method is easy to implement, but its disadvantages are the limitation of recognizable fonts (the first machine can read only ten fonts, while the number of fonts is increasing fast and fast), and the difficulty in dealing with non-ideal images (noisy, rotated, style-various). The second method, which can reduce the difficulty of the previous method, is feature-based method. In these methods, significant measurements are calculated and extracted from a character and compared to descriptions of the character classes obtained during a training phase. The description that matches most closely provides recognition. The features are given as numbers in a feature vector, and this feature vector is used to represent the symbol.

In this paper, we evaluate the accuracy of a convolutional neural network model implemented by Matconvnet library. The dataset includes both ideal and non-ideal images. Many groups of experiments are created to evaluate precisely and objectively. Because of limitation of time, in this paper, we only discuss about the classification and recognition step. The scanning, segmentation and preprocessing images are ignored, because these steps are relevant to other algorithms and techniques.

2. Litterature review

In ICPR 2014, a novel method of recognizing printed character is proposed. Instead of using square zoning, this method uses a more isotropic feature extraction regular hexagonal zoning. Combining gradient feature, a 2% improvement in accuracy was achieved. And the effectiveness of hexagonal zoning for recognition of high stroke count characters and lowresolution characters is confirmed by the experiments.

In the field of historical documents, a complete optical character recognition is proposed. This methodology consists of three steps: The first two steps refer to creating a database for training using a set of documents, while the third one refers to recognition of new document images. First, a pre-processing step that includes image binarization and enhancement takes place. At a second step, a top - down segmentation approach is used in order to detect text lines, words and characters. A clustering scheme is then adopted in order to group characters of similar shape. This is a semi-automatic procedure since the user is able to interact at any time in order to correct possible errors of clustering and assign an ASCII label. After this step, a database is created in order to be used for recognition. Finally, in the third step, for every new document image the above segmentation approach takes place while the recognition is based on the character database that has been produced at the previous step.

In natural scene text recognition, an automatic recognition method for color text characters extracted from scene images is proposed, which is robust to strong distortions, complex background, low resolution and non uniform lightning. Based on a specific architecture of convolutional neural networks, the proposed system automatically learns how to recognize characters without making any assumptions, without applying any preprocessing or post-processing and without using tunable parameters. Experimental results show an encouraging average recognition rate of 84.53%, ranging from 93.47% for clear images to 67.86% for seriously distorted images.

3. Method

3.1. Network structure

A convolutional neural network is more efficient than a fully-connected multilayer perceptron. First, with a multilayer perceptron, the system has to process with an extremely large trainable parameters. For example, a 24x24 input layer would already have 600 connections per single neuron in the hidden layer. By reducing the spatial resolution of the feature map, a certain degree of shift and distortion invariance is achieved [1-4]. Also, the number of free parameters is significantly decreased by using the same set of weights for all features in the feature map [1-1]. Secondly, convolutional neural networks are translation-invariant, due to the usage of convolutional filters, which are shifted through dimensions of the input. So CNNs can reduce the affect of shifting, scaling and other forms of distortion [1].

Because for this low number of free parameters, the training of convolutional neural networks requires far less computational effort than the training of multilayer perceptrons. This, as well as the implicit feature extraction and distortion invariance (to some degree), make convolutional networks an obvious candidate for classification tasks, especially pattern recognition. They have successfully been used for various pattern recognition tasks, such as handwriting and face recognition [1-1].

In this research, the authors use a common model of CNNs, LeNet-5 model. The input layer has a resolution of 32x32. The first convolutional layer has twenty feature maps, each of which has a resolution of 28 28, with a receptive field of 5 5. The second layer, or the first subsampling layer, contains twenty feature volumes of size 14 14x20. The third layer is another convolutional layer and has 50 feature volumes with size 1010x20, with a receptive field of 55. The fourth layer, or the second subsampling layer, contains 50 feature volumes as well, each of which is of size 5 5x20. The fifth layer is a convolutional layer with 500 feature maps, with a receptive field of 4x4x50. The first two subsampling layer use max pooling method, when the fifth layer, or the third subsampling layer uses relu method. The sixth layer is the last convolutional layer and has 26 feature volumes (corresponding to 26 lowercase English characters) with size 2x2x500. All neurons up to and including the eighth layer compute their input by calculating the weighted sum and feeding the result to the squashing function: softmaxloss.

(Picture1-CNN architecture used in this paper)

Dataset

The dataset is generated from 931 fonts downloaded from Google Fonts Projects, whose name is imdb and has the following structure:

(picture)

They are stored as the array. imdb.images.id is a 29,198-dimensional vector of numeric IDs for each of the 29,198 character images in the dataset. imdb.images.data contains 32x3232x32 images for each character, stored as a slide of a 32x32x29,198-dimensional array. imdb.images.label is a vector of image labels, denoting which one of the 26 possible characters it is. imdb.images.set is equal to 1 for each image that should be used to train the CNN and to 2 for each image that should be used for validation.

3.2. Training the model

The training function operates on Stochastic Gradient Descent (SGD) mini-batches of 100 elements, runs for 15 epochs and uses the learning rate of 0.001. In order to improve the network training speed, a momentum term can be added to the weight adjustment formulas [2]. In this paper, the chosen momentum factor is 0.9.

After training progression, the top1-error and top-5 error of training data is 0.0533 and 0.0065, respectively, and these results of validation data is 0.0738 and 0.0208, respectively.

(picture)

4. Experiment

The model is tested by a set of 32x32 images, each image includes only one character. The test set is divided into two main groups:

Ideal Image: The recognition accuracy is consistently 100% with ideal images (32x32 size, white-background, black-fill, non-noisy, non-rotated, gravity-center).

Non Ideal images: In this paper, the authors evaluate the recognition accuracy with these conditions:

1) Images with noise
2) Images with rotation
3) Images with non-center gravity
4) Images with thin stroke, compression

4.1. Images with noise

The noise in an image constitutes object with small area values (relative to the area of proper characters). 26 noisy images are created (we just add noise to ideal images).

23/26 images is well-recognized, which takes an accuracy of 88.46%. The three errors occur with l and v miss-recognized with y, and r miss-recognized with f.

(picture)

4.2. Images with rotation

With each of 26 characters, the authors test six different angles: -15, -10, -5, 5, 10, 15. From the result, we make the following observations:

•The rotated angle of 5 does not affect the recognition accuracy.

•The angle -5 and -10 have one error with m (miss-recognized with n), and the angle 10 has one error with l (miss-recognized with z).

(picture) miss-recognized with n

(picture) miss-recognized with z

•The angle -15 has 8 errors, but the angle 15 has only 3 errors. So we have a hypothesis that the clockwise rotation has less effect than the anti-clockwise rotation on the recognition accuracy.

4.3. Image with non-center gravity

Eight directions are tested, including North, South, East, West, NorthEast, NorthWest, SouthEast and SouthWest. Accuracy is 100% with North and South position, but unacceptable accuracy with other directions (¡ 50%).

We also combined this model with a simple segmentation to test with whole sentences. The sentences are represented on the 32xn ideal images, and the character is only in one line, with no uppercase and special characters. Fig [number] shows the example input image. The used segmentation is that determines white columns and separates the images into zones corresponding to these columns. A column is determined white if the ratio of sum of all pixel values on this column and sum of an absolute white column is greater than a threshold. In this paper, we choose this threshold is 0.97.

(picture)

Fig.[number] Example input sentence

We test with 10 sentences, each sentence is represented by 10 fonts. From the result, we have the following observations:

1. This method works well with fonts that characters are separated clearly, because after segmentation the problem returns to the recognition of ideal images.

2. (picture) t and h cannot be separated, and h is the recognized character.

(picture) r and a cannot be separated, and a is the recognized character.

(picture) o and g cannot be separated, and g is the recognized character.

(picture) r and l cannot be separated, and d is the recognized character.

We suppose that in case of failed segmentation, the bigger character, or the most similar character of this combination is recognized.

5. Conclusion

In this paper, we combine recognition individual character method by Matconvnet library with a simple segmentation to detect the whole sentences. By experimenting with two datasets, ideal images and non-ideal images from IMDB of Google, we proposal the best value of the threshold for segmentation and appreciate the accuracy in each character recognition. The result shown that the accuracy achieves up to 100% with ideal images, but just 88.46% with noisy images and less than 50% with East, West position of non-center gravity images. Therefore, non-ideal images recognition is a potential problem to research in the future.