社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

'Is quantity better than quality?'

—A Study Case on a Simple Classifier Applied to Digit Character Recognition—

Sabine BARRAT[†,††,†††,††††], Geoffrey ALARY[†,†††††], Masakazu IWAMURA[†], Koichi KISE[†], Seiichi UCHIDA[††††††], and Shinichiro OMACHI[†††††††]

† Graduate School of Engineering, Osaka Prefecture University, 1-1 Gakuencho, Naka, Sakai, 599-8531 Japan
†† Japan Society for the Promotion of Science
††††††† Graduate School of Engineering, Tohoku University
6-6-05 Aoba, Aramaki, Aoba, Sendai, 980-8579 Japan
†††††† Faculty of Information Science and Electrical Engineering, Kyushu University
744 Motooka, Nishi, Fukuoka, 819-0395 Japan
††††† EISTI Computer Science and Mathematics engineering school
avenue du Parc, 95011 Cergy-Pontoise Cedex, France
††† Nancy University - 24-30 rue lionnois - BP 60120 — 54 003 Nancy Cedex, France
†††† LORIA Lorraine Laboratory of IT Research and its Applications
Campus Scientifique - BP 239 - 54506 Vandoeuvre-l-Nancy Cedex, France
E-mail: †sabine.barrat@loria.fr, ††geoffrey@m.cs.osakafu-u.ac.jp, †††{masa,kise}@cs.osakafu-u.ac.jp,
††††uchida@ait.kyushu-u.ac.jp, †††††machi@ecei.tohoku.ac.jp

Abstract This paper deals with a learning method for digit character recognition. We propose to expand character image databases, by automatically generating deformations, noises and new fonts, and then to use a simple classifier based on k nearest neighbor technique. The results of first experiments show that the classifier is more robust on the resulting larger database. Moreover, performance on the MNIST digit character database is very promising.

Key words character recognition, MNIST database, k-NN

1. Introduction

Document image retrieval is still a very topical issue. In fact, document retrieval systems propose to access to documents containing large knowledge for different types of audiences.

The overall document image retrieval system can be divided in several steps: binarization, word/graphic segmentation, character extraction, feature extraction, matching/classification. In this paper, we focus on the last steps: feature extraction and classification, in order to recognize characters.

Classification is a basic task in data analysis and pattern recognition. This task requires a classifier, *i.e.* a function that assigns a class label to instances described by a set of features. The induction of classifiers from data sets of labeled data, or training sets (we speak about supervised learning) is a central problem in machine learning. Over the past few years, various methods have been proposed to deal with document image retrieval/classification. We can distinguish two main trends. The first one is based on the similarity concept. Such methods consist in allocating the query image to the class whose barycenter is the closest to the query image, by using a similarity or distance measure between image features. The k nearest neighbor algorithm (k-NN) belongs to this category of methods. These methods are well known for their implementation simplicity. Their performance is very dependent on the quantity of data and the robustness of features. They can be used with large training sets and large dimension features, but, in this case, they are very time and space consuming.

The second methods, more sophisticated, are called 'parametric learning methods'. It is the case of the approaches based on various functional representations such as decision trees, random forests, neural networks, decision graphs, associated to decision rules [1]. It is also the case of statistical methods like discriminant analysis, of the Support Vector Machines (SVM) or prob-

abilistic classifiers. All of these methods require the learning of parameters from the training set. Since, in the past, it was difficult to get and store large sets of labeled data, many approaches have been proposed in this direction, because they are known for their good performance in the presence of large dimension features, while needing a not so large number of data.

However, these last few years, thanks to the rapid growth of the Internet and multimedia information, large databases are now available and create a need in the development of image retrieval techniques, which can deal with such data masses. Nevertheless, sophisticated methods are often difficult to use when the size of the training set is large.

In order to overcome this problem, we propose an original method, using a simple nearest neighbor search and a large image training set. Actually, we enlarge a given training set, by automatically generating a lot of distortions on the original images and creating new examples for each class. By this way, we can easily obtain a large training set, more representative of the large variety of images. By this way and thanks to experiments done on the well-known MNIST database composed of $70,000$ handwritten digits ($60,000$ training images, $10,000$ test images), we will try to answer the question: 'Is quantity better than quality?', that is, is it better to use a large set of training data, associated to a simple classifier, or a sophisticated classifier with few data?

The organization of the paper is as follows. The Section 2 describes the process of database expansion. The simple features and classifiers we used to classify characters are described in Section 3. Section 4 presents the experimental results. Finally, conclusions and future works are given in Section 5.

2. Database expansion

The first step of the proposed method consists in automatically generating a large image database, by applying different kinds of deformations, with different intensities, to an existing image database, and by generating new models for each class. In this section, we present the distortions we used.

The set of chosen distortions is composed of two of the most common types of transformations in image processing: various affine transformations and dilations. Moreover, we used LaTeX fonts for generating new examples for each character.

2. 1 Affine transformations

An affine transformation is a very common 2-D geometric transformation which maps variables (*e.g.* pixel intensity values located at position (x_1, y_1) in an input image) into new variables (*e.g.* (x_2, y_2) in an output image) by applying a linear transformation and a translation. The linear transformation can be a geometric operation, or a linear combination of geometric operations: translation, rotation, scaling and/or shearing (*i.e.* non-uniform scaling in some directions).

In order to apply various affine transformations to our images, we used the affine transformation matrix

$$T = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In the last row of the matrix T, the two first values represent the translation. The 0 values mean we did not use translation. The sub matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ represents a linear transformation, which consists of a linear combination of 4 transformations. This linear transformation can be represented by a product of 4 matrix, each matrix representing a linear transformation:

$$A = \begin{bmatrix} \beta & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & -\tan\phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha & 0 \\ 0 & \frac{1}{\alpha} \end{bmatrix},$$

where:

- β is the scale parameter. The scale has a factor of β in the X-axis and Y-axis direction
- θ is the rotation angle
- ϕ is the shearing angle. The image is shearing in the X-axis direction with a shear angle of ϕ degrees. The angle is measured relative to a vertical Y-axis. The shearing widens the image by creating 'empty' triangles on the left or right side of the image
- α is a scale parameter: the scale has a factor of α in the X-axis direction, and a factor of $\frac{1}{\alpha}$ in the Y-axis direction

Let $X_{i,j}$ be the pixel intensity at the location (i, j) of the input image X. Each pixel of the image is then considered as a matrix 1×3: $\begin{bmatrix} i & j & 1 \end{bmatrix}$.

The transformed image O is obtained by mapping the pixel intensities of X to the new pixel locations $O_{i,j}$, where $O_{i,j}$ corresponds to the product: $\begin{bmatrix} i & j & 1 \end{bmatrix} T$.

Finally, this transformation requires the setting of 4 parameters: β, θ, ϕ and α. The parameter values we used in our experiments will be given in Section 4.. An example of an image on which we applied various affine transformations is given in Figure 1.

2. 2 Dilations

The dilation is a morphological transformation, based on the notion of intersection. A structuring element B is moved on an input image X, so that its center successively takes all the pixel coordinates of the image. For each position of B on the image X, if at least one pixel of B belongs to X, the center of B belongs to the dilated image. Then the dilation transformation requires two parameters: the shape of the structuring element (which can be a disk, a line, a rectangle, a square, ...) and the size of the structuring element.

The values of these parameters are given in section 4..

Finally, as an example, we show, in Figure 2, an image and some derived dilated images. These dilations were obtained with a disk as the structuring element. We can observe the effect of the size of the disk on the dilated images.

2. 3 LaTeX fonts

In order to generate new examples for each character, we used the LaTeX font catalogue (注1). In this catalogue, we chose a subset of the handwritten and calligraphical fonts. Moreover, for each font, we used all the styles (bold, italic, ...) available for this font. The default font for LaTeX is Knuth's Computer Modern, which gives default documents created with LaTeX. Changing the font, especially by using handwritten fonts, enables to provide original documents, in different styles, polices and which look like real handwritten documents, written by different writers.

In order to generate new examples for each character, we had to create a TeX file for each character and each font. Then each TeX file was compiled to get a pdf file. Finally each pdf file was converted in an image format.

The Figure 3 presents the 30 images we created thanks to LaTeX fonts, for the class of 1 digits. When it was possible, a same font was used several times, with different styles (italic, bold, ...). For each image, we give the name of the font we used for its creation.

3. Image representation and classification

'Is quantity better than quality?', that is to say, is a naive classifier with a large training set better than a sophisticated classifier trained with few data?

As a first step to answer this question, we chose to use the simplest descriptor and classifier as we can: the descriptor provides a vector containing the pixel intensities of the image, and the classifier is a simple k-NN we used with two different distance measures. The descriptor and the classifier, especially the distance measures we used, are briefly described below.

The descriptor we chose consists of the concatenation, line per line, of the pixel intensities of the image. That is to say, for a $m \times n$ image, we obtain a vector of $m \times n$ values.

Once the database is enlarged and the features are computed, we can classify the images. We chose to use a simple k nearest neighbor classifier (k-NN). Namely, for each query image, the set of the k nearest neighbors is built. The query image is allocated to the class which is the most represented in this set. Since the performance of this method is very dependent of the used similarity measure, we used two different measures of similarity: the classical euclidean distance, and the cosine similarity.

Let x and y be two feature vectors in an n dimension space. x and y have the same dimension and the coordinates of x are $x = (x_1, x_2, \ldots, x_n)$ and the ones of y are $y = (y_1, y_2, \ldots, y_n)$.

Then the euclidean distance $D1$ between x and y is given by:

$$D1(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

The cosine similarity $D2$, also known as Normalized Cross-Correlation (NCC), consists in measuring the angle between x and y, normed by their length (n), by using the cosine function:

$$D2(x, y) = \cos(x, y) = \frac{x.y}{|x|.|y|},$$

where $x.y$ represents the scalar product between x and y.

This similarity measure is often used in text retrieval, for measuring the semantic similarity between two documents. In this case, x and y represent the term frequency vectors of the documents. We chose this measure in addition to the euclidean distance because text retrieval methods applied to image retrieval have often shown good results, and this measure usually gives better recognition rates than the euclidean distance [2].

4. Experimental results

We used digit characters from the MNIST database [3] for our tests. This database of handwritten digits contains 60,000 handwritten digit images for the classifier training and 10,000 handwritten digit images for

(注1) : http://www.tug.dk/FontCatalogue

the classifier testing. Each image has a size of 28×28. This database has been chosen because there are many results for the MNIST database in the literature.

The first step of the experiments consisted of the expansion of the database. This process and the parameters settings are described below.

4. 1 Database expansion

In first, for each image of the MNIST training set, we applied 10 affine transformations. The parameter β was set to 1 and ϕ to 0. It means that we did not apply neither scale effect nor shearing on the images. The parameter α was set to 1.2, in order to apply an different scale effect on the X-axis and the Y-axis. Then, we changed the values of the parameter θ, corresponding to the rotation angle, in the range $\theta = \{-1.0, -0.8, \ldots, 0.8, 1.0\}$ (except 0 which corresponds to the identity rotation matrix) in order to generate rotated images (in the clockwise and the anticlockwise). This first transformations provided a new training set of $660,000$ images ($660,000$ comes from $60,000 + 10 \times 60,000$).

Then, we created a new set of 300 images by using handwritten and calligraphical LaTeX fonts. On each digit of this new set of 300 images, we applied various affine transformations. The scale parameter β was set to 1, and we changed the remaining three parameters α, ϕ and θ in the following ranges: $\alpha = \{1, 2\}$, $\phi = \{-1.0, -0.9, \ldots, 1.0\}$ and $\theta = \{-1.4, -1.2, \ldots, 1.4\}$. Thus, by combining them, we applied 630 affine transformations for each digit (630 comes from $2 \times 21 \times 15$). At this step, we obtained a new set of images based on LaTeX fonts, containing $189,000$ transformed images ($189,000$ comes from 630×300) and 300 original LaTeX images. Finally, on the set of the $189,000$ transformed images, we applied 3 dilations, with a disk as the structuring element. The size of the disk was changed in the range $\{1.0, 1.5, 2.0\}$. Thereby, we obtained $567,000$ new affinely transformed and dilated images ($567,000$ comes from $189,000 \times 3$).

Therefore, at the end of the database expansion, we disposed of a large database of $1,416,300$ images decomposed in 5 subsets:

- dataset 1: initial MNIST database ($60,000$ images)
- dataset 2: images of MNIST database with affine transformations ($600,000$)
- dataset 3: images created thanks to LaTex fonts (300 images)
- dataset 4: images based on LaTeX fonts transformed with affine transformations but without dilation ($189,000$ images)
- dataset 5: images based on LaTex fonts transformed with affine transformations and dilation ($567,000$ images)

Finally, all the images of the database were normalized, by applying trimming and by resizing the images to their original size (28×28). Here, the trimming consist in subtracting the pixels of the image border in order to keep the smallest area which contains the digit.

4. 2 Experiments and results

All the experiments were done by taking, successively, each image of the MNIST test set (composed of $10,000$ images) as a query image. Different subsets described above, and combinations of these subsets, were successively used as training set, in order to evaluate the contribution of the different image transformations and the size of the training set, on the recognition rate. For all the experiments, the k-NN classifier were applied for the two distance measures and different values of k: $k = \{1, 3, 5\}$.

The Figure 5 (respectively 4) shows the recognition rate, in function of different training sets and different values of k, obtained with the euclidean distance (respectively cosine similarity (NCC)). The rates obtained for $k = 1$ are represented by the first bars, the ones obtained for $k = 3$ are represented by the second bars and the third bars represent the performance obtained for $k = 5$. On the X-axis, we can see the different training sets we used, from the initial MNIST database, until the whole expanded database.

Since the best results are usually obtained for the NCC similarity, and this distance is also known to be more stable than the euclidean one, we will focus on the comments of the Figure 4.

We can observe the impact of the k value on the recognition rate: the best recognitions rates are usually obtained for $k = 5$. Besides, we can see the effect of the different transformations on the recognition rate: the generation of images with various affine transformations increases the recognition rate (see the difference between the two first bars, datasets 1 et 2). This means that the affine transformations provided us with a more complete training set, more representative of the handwritten characters.

On the contrary, using the images created thanks to LaTeX fonts does not have a positive effect on the recognition rate (see the difference between the second and the third bar). The bad contribution of the images

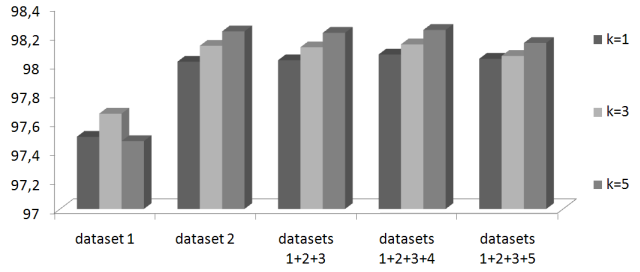

Fig. 4 Recognition rate in function of the training set and the k value - NCC similarity

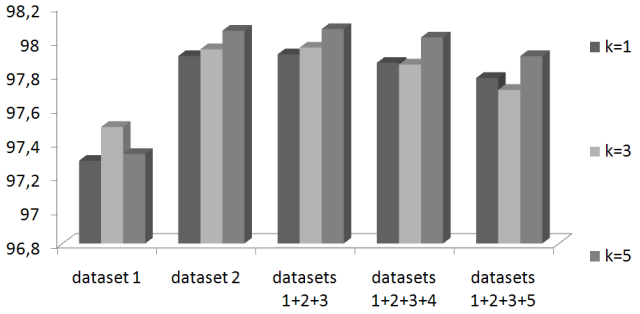

Fig. 5 Recognition rate in function of the training set and the k value - euclidean distance

based on LaTeX fonts is probably due to the gap between the images of the original MNIST training set, which are real handwritten characters, and the images obtained with LaTeX which are artificial handwritten characters. That is, the images obtained thanks to LaTeX fonts seem to be to far from the original MNIST database.

Finally, we can see that the application of other transformations on the images obtained with LaTeX (application of affine transformations in datasets $1+2+3+4$, and application of dilations in datasets $1+2+3+5$) leads to a recognition rate decrease. Particularly, the dilation slightly deteriorates the results, because the resulting dilated images are too distorted and then make confusion between classes. For example, a too noisy 7 digit can be closer from a 1.

5. Conclusions and future works

In this paper we proposed an original classification method which consists in using a naive classifier, associated to a large training set. The large database was obtained by automatically extending an existing training set. To expand this training set, we used several commons geometric and/or morphological transformations and we created new examples for each character thanks to the LaTeX font catalogue. Many experiments were done on the well-known MNIST database. The results show that the classifier performance is improved by using a larger database, as shows the addition, to the training set, of affine transformed images. Nevertheless, the expansion of the database should be reconsidered: we have to take care to the distortion level of the generated images. In fact, if the generated images are closer from an other class, the addition of such images in the training set would make the recognition rate decrease.

Finally, future works will be dedicated to deal with large databases with a reasonable time and space complexity.

Acknowledgment

This work was supported in part by the JSPS Postdoctoral Fellowship for North American and European Researchers (Short-Term) and Grant-in-Aid for Scientific Research (B)(22300062) from JSPS.

References

[1] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[2] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, Second Edition, Wiley-Interscience, 2001.

[3] Y. Lecun and C. Cortes, "The mnist database of handwritten digits `http://yann.lecun.com/exdb/mnist/`,"

original image deformed images with affine transformations

Fig. 1 Example of an image on which we applied various affine transformations

original image disk 1.0 disk 1.5 disk 2.0

Fig. 2 Example of an image on which we applied three different dilations

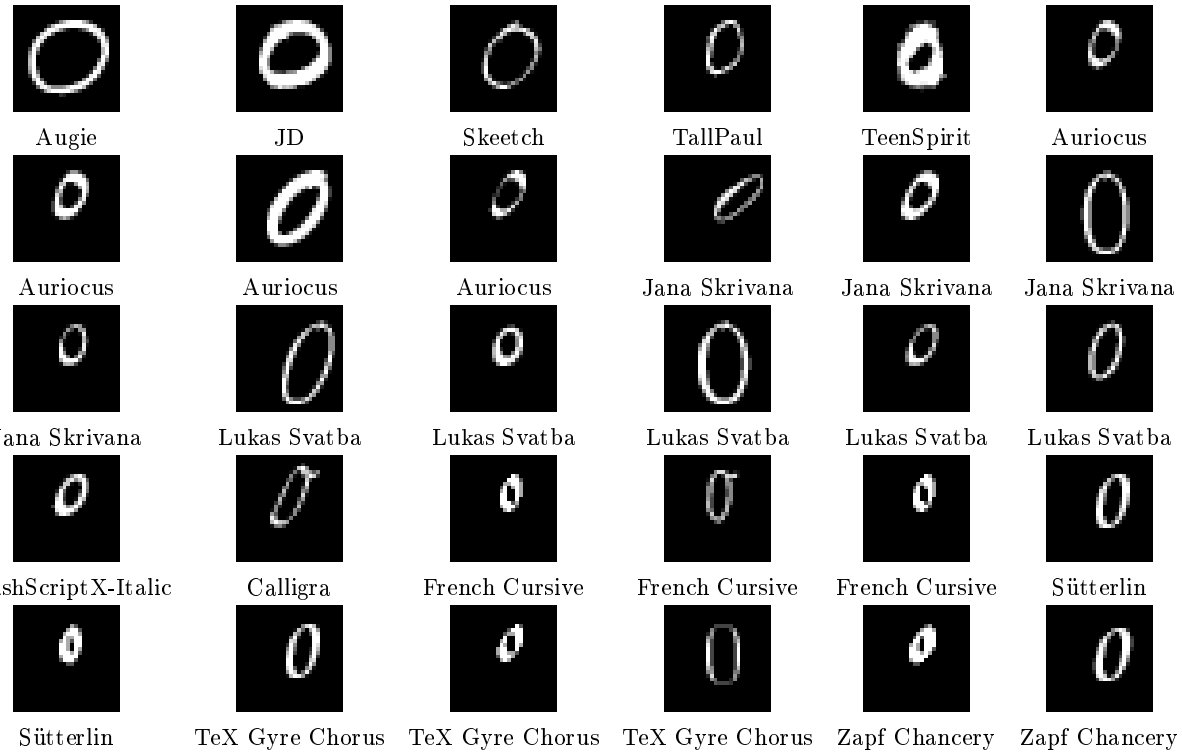

Fig. 3 Examples of images created with LaTeX fonts, and the corresponding Fonts