Optical Font Recognition for Multi-Font OCR and Document Processing

Serena La Manna** Anna Maria Colla* Alessandro Sperduti**
*Elsag S.p.A., Via G.Puccini,2, 16154 Genova, Italy
*e-mail: Annamaria.Colla@elsag.it
**Dipartimento di Informatica, Università di Pisa
**Corso Italia, 40, 56125, Pisa, Italy
**e-mail: {lamanna,perso}@di.unipi.it

Abstract

In this paper we present a Multi-font OCR system to be employed for document processing, which performs, at the same time, both the character recognition and the font-style detection of the digits belonging to a subset of the existing fonts. The detection of the font-style of the document words can guide a rough automatic classification of documents, and can also be used to improve the character recognition.

The system uses the tangent distance as a classification function in a nearest neighbour approach. We have to discriminate among different digits and, for the same character, we have to discriminate among different font-styles. The nearest neighbour approach is always able to recognize the digit, but the performance in font detection is not optimal. To improve the performance of the system, we have used a discriminant model, the TD-Neuron, which is employed to discriminate between two similar classes. Some experimental results and prospective use in document processing applications are presented.

1. Introduction

Document processing is a complex task, consisting of several steps and employing different techniques according to its specific purpose. In general, the first step of document processing is the acquisition process, where an optical scanner is used to acquire *paper documents*. The last step is indexing, with the purpose to extract appropriate keys to store with each document for subsequent retrieval. For most documents, belonging to different domains in several application environments, indexing requires an *Optical Character Recognition (OCR)* step, to provide the translation of human-readable characters into machine-readable codes. OCR systems aim to give a rapid and automatic method to store documents in a computer: indexes are text-based, from keywords to natural language sentences, and can be used to retrieve documents whose text content has been read and stored in a database.

If we consider *machine-printed documents*, we can divide the OCR systems in three groups: Mono-font, Multi-font, and Omni-font. *Mono-font* OCR systems deal with documents written with one specific font: their accuracy is very high but they need a specific module for each font. *Omni-font* OCR systems allow the recognition of characters of any font, and for this reason their accuracy is typically lower. Finally, *Multi-font* OCR systems handle a subset of the existing fonts. Their accuracy is related to the number and the similarity of the fonts under consideration. These systems achieve the best results when a single letter has very similar features in each font and it is easy to discriminate among different classes. On the other hand, the recognition is very difficult when different letters have similar features: for example the letter 'l' in one font could be very similar to the digit '1' in another font. There are several methods to get over this kind of problem, such as the use of a context dependent post-processing to distinguish between letters and digits [1] [2] or the use of an *Optical Font Recognizer (OFR)*, to detect the font type and subsequently convert the multi-font problem into mono-font character recognition. An OFR can be useful also to simply characterize single characters, words or paragraphs in a printed document, as an aid to analysis of document characteristics and layout.

The paper is organized as follows. In Section 2 we introduce the problem of Optical Font Recognition and propose a solution based on Tangent Distance Techniques. More details about the Tangent Distance, Tangent Models and TD-Neuron are supplied in Section 3. In Section 4 we specify the problems met during the acquisition process and the pre-processing choices we made. Some experimental results are reported in Section 5, where we present a comparison between a 1-NN classifier, based on the Euclidean Distance, and a 1-NN classifier based on the one-sided Tangent Distance. In the same Section we discuss the results obtained by the TD-Neuron for the treatment of the above-mentioned

difficult cases. Finally, the conclusions are reported in Section 6.

2. Optical Font Recognition for document processing

Optical Font Recognition (OFR), i.e the detection of the font style of the documents, can be useful for:

- document characterization / classification;
- document layout analysis;
- improvement to Multi-font OCR.

If the font is a specific document feature, different kinds of documents can be distinguished by their font-style and the font-style detection can guide a rough automatic classification of documents. So this information can help in addressing different documents to different processing.

Often the font-style is not the same for a whole document: in these cases the font is a word feature, rather than a document feature, and its detection can be used to discriminate between different regions of the document, such as title, figure caption or normal text. The detection of the font-style of a word can also be used to improve character recognition: we know that Mono-font OCRs achieve better results than Multi-font ones, so the recognition of a document can be done using first an OFR, and then a Mono-font OCR.

The OFR problem has been often underestimated, in spite of its usefulness for document processing, so there is little literature about it. In [3], Zramdini presents a font recognition system which performs the font detection without any knowledge of the characters in the documents. The results are very interesting, since the recognition rate is near 96%, but the system does not perform any character recognition.

Our approach is different, since we carry out the *simultaneous recognition of both font and letter* of any character in a document. By doing this, the system can translate the papers by OCR and discriminate among different kinds of documents at the same time. Our aim was to develop a system able to process images acquired at a low resolution level (200 dpi) and invariant to some specific transformations of the input patterns such as small rotations and/or location shifts. An external module was used to perform the segmentation of the documents and then the character images were given as input to the system, which had to label them with the correct font and letter. We wanted the system to be transformation invariant, so we decided to use, as the classification distance, the *one-sided tangent distance*, a particular version of the *tangent distance*, introduced by Simard in [4].

The one-sided tangent distance, used with the *tangent model* [5], yields satisfactory results on characters belonging to fonts with discriminant features. In other difficult cases, when a character in a specific font is very similar to the same character in another font, the system identifies the input pattern as a member of the set constituted by the two similar classes. In these cases, to improve the performance, the output of the classifier is given as input to another model, the *TD-Neuron*.

3. Tangent Distance Overview

Consider a pattern $\boldsymbol{X}_i$ and a set of n transformation $\boldsymbol{\theta}$. The function $\boldsymbol{X}_i(\boldsymbol{\theta})$ is a manifold of dimension at most n, where $\boldsymbol{X}_i = \boldsymbol{X}_i(0)$. Since the computation of the distance between two manifolds is a very hard problem, Simard et al. [4] proposed a linear local approximation of the manifold by its tangent subspace at the point $\boldsymbol{X}_i$:

$$\tilde{\boldsymbol{X}}_i(\boldsymbol{\theta}) = \boldsymbol{X}_i + \sum_{j=1}^{n} \boldsymbol{T}^j_{X_i} \theta_j \,,$$

where $\boldsymbol{T}^j_{X_i}$ are n different tangent vectors at the point $\boldsymbol{X}_i(\boldsymbol{\theta})$, which can easily be computed by finite difference. This approximation is accurate only for local transformations, however, global invariance, in the case of Character Recognition, may not be desired, since it can cause confusion between patterns such as “9” and “6”. The distance between two manifolds can be approximated by the distance between their associated subspaces, called *Tangent Distance*.

Two versions of *tangent distance*, with decreasing complexity, have been proposed in the literature. The first, defined between two subspaces, is called *two-sided tangent distance*:

$$D_T(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{\alpha},\boldsymbol{\theta}} \|\tilde{\boldsymbol{X}}_i(\boldsymbol{\alpha}) - \tilde{\boldsymbol{X}}_j(\boldsymbol{\theta})\|. \qquad (1)$$

while the second version, defined between a pattern and a subspace, is called *one-sided tangent distance* [6]:

$$D_T^{\text{1-sided}}(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{\alpha}} \|\tilde{\boldsymbol{X}}_i(\boldsymbol{\alpha}) - \boldsymbol{X}_j\|. \qquad (2)$$

In the case of the *one-sided* version of the tangent distance, the *tangent model* for a class C is defined by the equation:

$$\boldsymbol{M}_C = \arg\min_{\boldsymbol{M}} \sum_{p=1}^{N_C} \min_{\boldsymbol{\theta}_p} \|\boldsymbol{M}(\boldsymbol{\theta}_p) - \boldsymbol{X}_p\|^2, \qquad (3)$$

which can be easily solved by Principal Component Analysis theory. This model is non-discriminant, since it is generated using patterns belonging to a single class. The training of a new class requires only patterns belonging to

that specific class and does not compromise the existing data, so non-discriminant models are very useful when a simple expansion is required. On the other hand, the knowledge about the differences among classes is not stored in the system and its lack can reduce the classification accuracy, especially when different classes have very similar features. In such cases it would be better to use a discriminant model, which identifies each class with its own features and with its differences with respect to the other classes. The resulting system is very difficult to expand so we have restricted the use of the discriminant model to distinguish between two classes with very similar features.

Sona et al. [7] proposed a gradient descent constructive algorithm, the *TD-Neuron*, that develops discriminant tangent models. A *TD-Neuron* is characterized by a set of internal weights which determine a tangent model. This set of parameters is composed by $n + 1$ vectors, including a *centroid* $\boldsymbol{W}$ and n tangent vectors $\boldsymbol{T}_i$ which constitute an orthonormal basis. The neuron is trained with a gradient descent approach on the error function and the usual Mean Square Error, using a constructive algorithm.

4. Acquisition and Pre-processing

Our goal was to implement a system able to perform *both character recognition and font detection simultaneously*. The training and the testing of the system required a database of characters labeled with the associated font and letter. We were not able to find a public domain database organized according to these features, so we had to create our own database. The OFR problem is quite easy to solve when different fonts have discriminant features. On the other hand, it is harder to detect the right character font when different classes have very similar characteristics (see Figure 1). In our experiments 8 fonts were considered: 4 with original features, Arial Narrow, Comic Sans Serif, Impact, Verdana, and 2 couples of fonts with similar features, i.e. Arial, Lucida Sans Unicode, Lucida Console and Tahoma.

A preliminary test was carried out using digits only. The training set was created using also lower- and uppercase letters, to evaluate, in the test phase, the recognition of digits such as '8', '1', '0', which are often confused, by multi-font OCR, with letters such as 'B', 'l', 'I', 'O'. For each font, four documents were created, three for the training set, containing 60 examples of each character (letters and digits), and one for the test set, containing 40 instances of each digit. To evaluate the performance of the system on images of imperfect quality and to maintain a fast and low-cost acquisition, the documents, reproduced with a laser printer, were acquired at a low resolution, 200 dpi, in gray levels.

The images acquired were then binarized, using commercial tools, to perform the document segmentation using a simple OCR (developed at Elsag's Research lab). The used OCR was not multi-font, so, during the process, many images were lost. Therefore the number of images in the training set and in the test set is lower than the number of characters in the documents, especially for the classes "1" and "7". The images given as output by the OCR were perfectly cut, so they were embedded in a white background, with a 40x40 format, to obtain images of the same size. The binary images were then transformed into gray level ones (see Figure 2), using a mean weighted operator, to exploit the peculiarities of the tangent distance, which deals with continuous values.

5. OCR and OFR Results

A 1-Nearest Neighbour classifier with the *Euclidean Distance* was considered to evaluate the difficulties for the OFR problem. The prototype for each class was the *centroid* calculated as the mean value of the input patterns. During the test the system computed, for each input pattern, the Euclidean Distance between the pattern and each prototype. The test was carried out on two similar fonts, Arial and Lucida Sans Unicode, to observe the behaviour of the classifier in a difficult case. The results are reported in Table (1). It can be easily noted that the 1-NN classifier based on the Euclidean distance is not able to reach a satisfactory performance.

The second test consisted in implementing a 1-Nearest Neighbour classifier based on the *two-sided tangent distance*. This distance allows to incorporate an a priori knowledge about the pattern transformations, so we decided to consider rotations by a minimum angle (up to 5% of π) and

Figure 1. *Examples of characters with similar features; the font-styles are, left to right: Tahoma, Lucida Console, Arial, Lucida Sans Unicode.*

Figure 2. *Images at the end of pre-processing: a '2 Arial' (left) and a '2 Lucida Sans Unicode' (right). The two images have very similar features.*

Table 1. Number of correctly recognized characters of two fonts, Arial and Lucida Sans Unicode, using Euclidean 1-Nearest Neighbour algorithm. The test set contains 40 examples of each class, excluding class '7' Arial (12 examples), and class '7' Lucida Sans Unicode (17 examples).

FONTS	ARIAL	LUCIDA S.U.
0	26	21
1	**3**	37
2	17	33
3	10	30
4	24	24
5	13	29
6	18	27
7	8	14
8	16	29
9	18	25

shifts of a fixed number of pixels (up to 10% of the character height and the same for the width). The number of relevant tangent vectors, i.e. the tangent subspace dimension, for each class was empirically determined as 6. However, the two-sided tangent distance is too computationally expensive for solving the OFR problem, so we considered the *one-sided* version of tangent distance, which is easier to compute. Furthermore recent works [8] have shown that the *one-sided tangent distance* may give, in some cases, better results than the *two-sided* version. So a 1-NN classifier based on the one-sided tangent distance was implemented. Again, the tangent subspace dimension was empirically chosen equal to 6.

The results of the classification were very interesting: the characters were always recognized (recognition rate = 100%) in spite of the low resolution. The font-detection rate was very different for the two above-mentioned font groups. For the fonts with particular features, the font detection rate was better than 98%, since the system obtained 100% for Comic Sans Serif and Impact, 99,75% for Arial Narrow (one misclassified digit in class "1") and 98,9% for Verdana (three misclassified digits in class "1" and one misclassified digit in class "7"). For the couples of fonts with similar features, the results were not so positive: the classifier identified 2 ambiguity groups corresponding to the couples of above-mentioned similar fonts. The characters were classified in the correct ambiguity group but the classifier could not determine the exact class. The results of the classification for these fonts are reported in Table (2). The classification rate for the fonts with similar features is not high, but we have to underline that the experiment concerned font discrimination on single characters (word-based discrimination would be easier). However, we tried to improve the recognition rate for the fonts in the ambiguity groups, using a discriminant model, the *TD-Neuron*. Using a TD-Neuron for each class in our problem is not suitable, as we have previously discussed, but the discriminant models could be useful in specific, difficult cases, such as the ambiguity groups.

Table 2. Classification of characters Arial, Lucida Console, Lucida Sans Unicode, Tahoma. The label of the columns is the output of the classifier.

	Arial	Lucida Console	Lucida S.U.	Tahoma	Verdana
Arial	160	0	212	0	0
Lucida C.	0	105	1	271	3
Lucida S.U.	38	0	338	1	0
Tahoma	0	132	0	253	0

The test was carried out considering the ambiguity group (Arial, Lucida Sans Unicode), and implementing one TD-Neuron for each digit. In this way each neuron had to discriminate between two classes, i.e. the same digit in the two fonts. The results of the classifier were given as input to the correct TD-Neuron, since the characters were always correctly recognized. The first test we carried out produced no satisfactory results, since the best recognition rate was about 65%. These unsatisfactory results were due to the fact that the characters were embedded in a white background and the number of white pixels was often higher than the number of gray ones. So the pixels of the characters had a lower influence in the computation, than the pixels of the background. The gray values of the pictures were then reverted, and the neurons were trained all over again.

The results, reported in Table (3), are positive, especially for the Lucida Sans Unicode characters, since the recognition rate is between 85%, obtained on class "1", and 100%, obtained on three classes, "5", "6", and "7".

Table 3. Best results of the TD Neurons.

Class	Iterations	# Tangents	Arial Errors	Lucida S.U. Errors
0	3169	12	18	5
1	5548	12	17	6
2	7023	12	20	2
3	5024	12	31	1
4	5929	12	19	1
5	5771	12	31	**0**
6	5297	12	22	**0**
7	4904	4	9	**0**
8	7383	12	24	1
9	4973	12	22	2

For Arial characters the recognition rate is lower, however, the performance of the system is very interesting, especially if we consider that the detection of the font for practical purposes should be bound to an entire word and not to

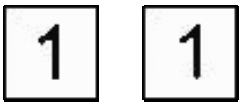

Figure 3. *Examples of misclassified digits: a '1 Arial' classified as '1 Lucida Sans Unicode' (left) and a '1 Lucida Sans Unicode' classified as '1 Arial' (right).*

a single character. From this point of view, we will perform font detection using a majority criterion on the characters of an entire word. Results must also be analyzed on the ground of the low quality of the images, that sometimes did not allow the font detection even by a human expert (see Figure 3). This low quality is due to the rough pre-processing, adopted to guarantee a fast and low-cost acquisition, and low memory effort.

6. Conclusions

In this paper we have presented a model, which uses the Tangent Distance as a classification distance in a Nearest Neighbour approach, capable to perform OFR and (Multifont) OCR simultaneously. This model is applicable to several document processing applications. The classes in our problem were the single characters in the 8 fonts under consideration: each class is represented by a prototype, the *tangent subspace model*. The training of the system was carried out using lowercase letters, uppercase letters and digits of the considered fonts, while the test of the system has been done on digits only.

Using non-discriminant models, the best results were given by a 1-Nearest Neighbour classifier, based on the *one-sided tangent distance*. The recognition rate for characters was 100% and the detection of the font yielded interesting results, especially for fonts with specific features. In difficult cases, when 2 fonts had very similar features, we used a discriminant model, the *TD-Neuron*, which was able to improve the performance of the classifier, reaching satisfactory results.

Our work has proved that the tangent distance is suitable for character recognition and font detection. The implementation is currently under study: we would like to develop a system able to produce, given a document, a text file with the recognized characters and fonts. In particular, this "OFR+OCR" module could be employed to perform some document processing operations aimed at layout analysis and document characterization/classification. Next steps to complete the OFR module are: (i) the implementation of a majority filter to recognize the "preeminent" font in a written word (or paragraph), which, besides improving accuracy, would be useful both for layout analysis and document characterization, and (ii) the analysis of emphasis indicators, such as boldface or italics, and font size.

Since it is very important to obtain a fast processing of the documents, we will also try to speed up the system. In fact, the number of the classes in our problem is very high, and it takes a long time to recognize the right font and character of each input pattern. The main idea is to reduce the number and the computational efforts of the distances to be computed for each step, creating a hierarchy of distances and a hierarchy of image resolution levels, as proposed by Simard in [9].

References

[1] J. Wang and J. Jean. Resolving multicharacter confusion with neural networks. *Pattern Recognition*, 26(1):175–187, 1993.

[2] F.Lebourgeois and J.L.Henry. A contextual processing for an ocr system, based on pattern learning. In *2nd Int'l conference document analysis and recognition*, pages 862–865, Los Alamitos, Calif., 1993. CS Press.

[3] A. Zramdini. *Study of optical font recognition based on global typographical features*. PhD thesis, University of Fribourg, 1995.

[4] P. Y. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 50–58, San Mateo CA, 1993. Morgan Kaufmann.

[5] T. Hastie, P. Y. Simard, and E.Säckinger. Learning prototype models for tangent distance. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 999–1006, Cambridge MA, 1995. MIT Press.

[6] H. Schwenk and M. Milgram. Transformation invariant autoassociation with application to handwritten character recognition. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 991–998, Cambridge MA, 1995. MIT Press.

[7] D. Sona, A. Sperduti, and A. Starita. A constructive learning algorithm for discriminant tangent models. In Michael I.Jordan Michael C.Mozer and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 786–792, Cambridge MA, 1997. MIT Press.

[8] D. Sona, A. Sperduti, and A. Starita. Discriminant pattern recognition using transformation invariant neurons. To appear on Neural Computation.

[9] P. Y. Simard. Efficient computation of complex distance metrics using hierarchical filtering. In J.D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 168–175, San Francisco CA, 1994. Morgan Kaufmann.