Pattern Recognition 41 (2008) 3044 – 3053

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Filtering segmentation cuts for digit string recognition

E. Vellasques[a], L.S. Oliveira[a,*], A.S. Britto Jr.[a], A.L. Koerich[a], R. Sabourin[b]

[a]Pontifical Catholic University of Parana (PUCPR), Curitiba, Brazil R. Imaculada Conceição, 1155, 80215-901 Curitiba, PR, Brazil
[b]Ecole de Technologie Superieure, 1100 rue Notre Dame Ouest, Montreal, Quebec, Canada

ARTICLE INFO

Article history:
Received 12 January 2007
Received in revised form 20 December 2007
Accepted 13 March 2008

Keywords:
Handwriting recognition
Segmentation
Filtering

ABSTRACT

In this paper we propose a method to evaluate segmentation cuts for handwritten touching digits. The idea of this method is to work as a filter in segmentation-based recognition system. This kind of system usually rely on over-segmentation methods, where several segmentation hypotheses are created for each touching group of digits and then assessed by a general-purpose classifier. The novelty of the proposed methodology lies in the fact that unnecessary segmentation cuts can be identified without any attempt of classification by a general-purpose classifier, reducing the number of paths in a segmentation graph, what can consequently lead to a reduction in computational cost. An cost-based approach using ROC (receiver operating characteristics) was deployed to optimize the filter. Experimental results show that the filter can eliminate up to 83% of the unnecessary segmentation hypothesis and increase the overall performance of the system.

1. Introduction

The recognition of unconstrained handwritten numerals is still a very active area of research. It is composed of several steps, including image acquisition, pre-processing, segmentation, representation, and recognition [1]. Segmentation is a very challenging task as we need to "split" two or more digits so that they can be later recognized by a general-purpose classifier; but we also need to know what we are segmenting and that involves some recognition. Early methods used to make heavy use of constraints on document format in order to reduce segmentation complexity. On first generation OCR's, due to memory limitation, each character had to be scanned individually before its recognition. This approach required pre-scans, where the positions of the characters to be recognized had to be detected first, with the use of reference marks [2].

There are two main tasks in segmentation. The first is connected component detection. Through connected component detection, all the elements are identified. These elements can be isolated digits, broken parts of digits, delimiters, and touching digits. Usually some post-processing is added to this task so broken parts can be grouped. The second and most challenging task is the segmentation of touching digits. A connection between two digits occurs when their foreground pixels merge, creating a bigger connected component. There are two major categories of touching numeral strings, single- and multiple-touching [3]. Fig. 1 shows the most common types of touching.

Casey and Lecolinet [1] proposed a taxonomy for segmentation strategies. According to them, the segmentation strategies can be found in an orthogonal space with three axes, namely recognition-based, holistic, and dissection. Usually, recognition-based methods make less use of heuristics. However, they usually generate too many segmentation hypotheses, and it can become a bottleneck as each digit of these hypotheses has to be later verified by a general-purpose classifier. The dissection methods, otherwise, usually generate less segmentation hypotheses, but depend heavily on heuristics. The literature has many examples that show this taxonomy [4–6].

In Fujisawa et al. [2], heuristics are avoided but an average of three segmentation points are found for each two-digit string. In Chen and Wang [3], the use of heuristics is also avoided, but in this case, the average of points found for each two-digit touching string is 7.3 [7]. Fig. 2a gives an insight of how many hypotheses should be evaluated by the classifier due to over-segmentation. In this case, suppose that an algorithm proposes SP_0, SP_1, and SP_2 as segmentation cuts, which will divide the image in four segments (C_0, C_1, C_2, and C_3). The set of possible segmentation hypotheses can be represented by a graph, where each segment is represented by a vertex and each segmentation cut is represented by an edge in this graph. SP_1 is the optimal segmentation cut, while SP_0 and SP_2 are unnecessary cuts generated due to the over-segmentation nature of the segmentation algorithm.

A segmentation cut is an "incision" which is applied to a given section of a stroke, which splits the connected component in two parts. Since a connected component may contain more than one character and the connection can occur in more than one region, a common approach is to apply more than one "incision". Each segmentation cut can be turned "on" and "off" accordingly.

* Corresponding author. Tel.: +55 41 3271 1348; fax: +55 41 3271 2121.
E-mail address: soares@ppgia.pucpr.br (L.S. Oliveira).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.03.019

The combination of segmentation cuts and their states (''on'' and ''off'') for a given connected component will determine the set of segmentation hypotheses. A segmentation hypothesis can be seen though as one of the 2^N possible states, where N is the number of segmentation cuts. A segment will be one of the resulting pieces of the connected component after the ''incisions'' corresponding to a given state have been applied. Here, a segmentation hypothesis can also be seen as the union of such segments. For example, on Fig. 2, if all segmentation points where turned ''on'', it would result in four segments. This segmentation hypothesis is represented by the central path of the depicted graph. In Fig. 2, the optimal hypothesis is obtained when SP_1 is turned ''on'' and SP_0 and SP_2 are turned ''off''. However, turning SP_1 and SP_2 ''on'' and SP_0 ''off'' will also result in a high recognition score, in the system proposed by Oliveira et al. [8], leading to recognition error.

The segmentation hypotheses can be generated by several different ways. In the case of Fig. 2a we simulated a classic segmentation algorithm that uses peaks and valleys of the contour to define segmentation cuts.

Finding optimal segmentation cuts in a straightforward and general manner is something very difficult due to variability in the location of segmentation cuts. Furthermore, some results of over-segmentation can be easily confused with an isolated digit (e.g., Fig. 2b where those pieces can be confused with the digits ''1'' and ''0''). In the example depicted in Fig. 2a, the path ''510'' may produce a higher score than the path ''56''. This kind of problem makes the use of heuristics to some extent necessary. In this context, a challenging problem lies in reducing the use of heuristics without increasing the number of segmentation hypotheses or vice versa due to the lack of general rules to describe points along with the variability of points location.

Instead of creating a new segmentation method without both heuristics and over-segmentation, in this paper we propose a novel approach to reduce the number of segmentation hypotheses in a cost effective manner. We implemented this through the use of a filter, placed between the segmentation and recognition modules. The purpose of this method is to classify the segmentation cut into necessary or unnecessary prior to any attempt of recognition by a general-purpose classifier, what would cause a reduction in the complexity of the graph shown in Fig. 2a, and could consequently reduce the computational cost. Since we are dealing with a 2-class problem (necessary and unnecessary segmentation cuts), as explained in Ref. [9] we have chosen SVM [10] in order to model the segmentation cuts.

The draft of this concept was first proposed by the authors in Ref. [9] and to the best of our knowledge, there is no similar method in the literature. The main idea behind the proposed method is that unnecessary segmentation cuts are modeled through the use of over-segmented digits, rather than trying to model unnecessary cuts through their structural features. A cost-based approach using ROC (receiver operating characteristics) [11] was deployed to optimize the proposed filter. We have performed experiments using different segmentation algorithms to demonstrate the impacts of such a filter. Experimental results show that the filter can eliminate up to 83% of the unnecessary segmentation hypotheses. We also show that the ROC-based cost mechanism increases the overall performance of the system.

The remaining of this paper is organized as follows: Section 2 describes the baseline system we have used in our experiments and introduces the architectures based on verification and filtering. Section 3 describes the feature set used to train such a filter and also presents the basics about ROC. Section 4 reports the experiments we have performed and Section 5 concludes this work.

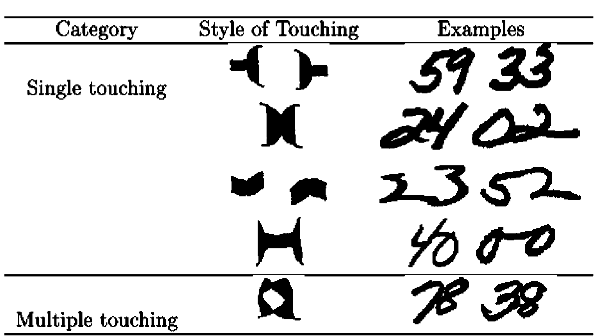

Fig. 1. Types of touching between numerals [3].

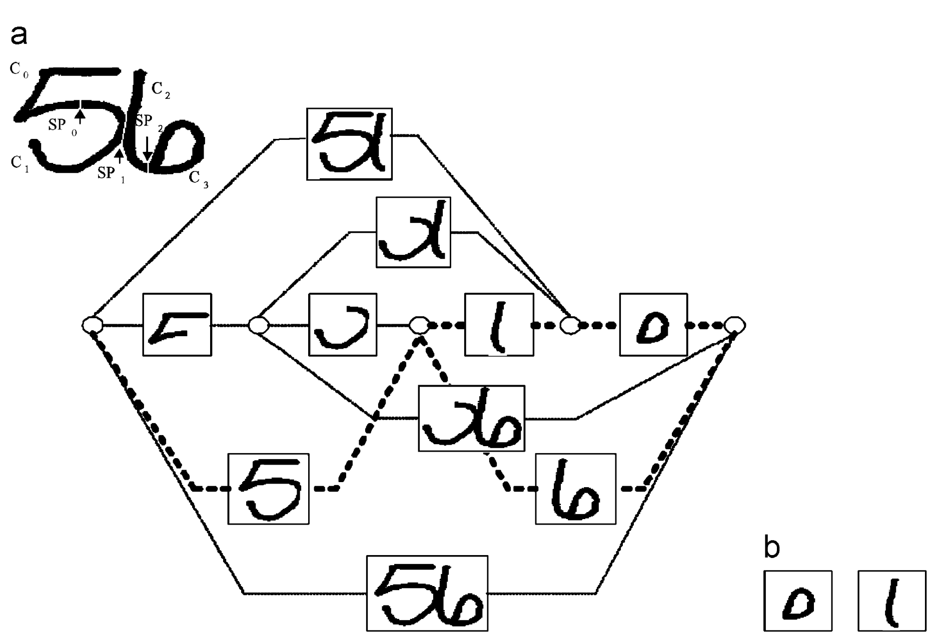

Fig. 2. (a) Segmentation paths for the string ''56'' and (b) images that can be easily confused with digits ''1'' and ''0''.