The image Text Recognition Graph (iTRG)

Zohra Saidane[1], Christophe Garcia[1] and Jean Luc Dugelay[2]
(1) Orange Labs - 4, rue du Clos Courtel 35512 Cesson Sévigné Cedex, France
firstname.lastname@orange-ftgroup.com
(2) Eurecom - 2229 route des Crete 06904 Sophia Antipolis, France
Jean-Luc.Dugelay@eurecom.fr

Abstract

This paper presents a graph based scheme for color text recognition in images and videos, which is particularly robust to complex background, low resolution or video coding artifacts. This scheme is based on a novel method named the image Text Recognition Graph (iTRG) composed of five main modules: an image text segmentation module, a graph connection builder module, a character recognition module, a graph weight calculator module and an optimal path search module. The first two modules are based on convolutional neural networks so that the proposed system automatically learns how to robustly perform segmentation and recognition. The proposed method is evaluated on the public ICDAR 2003 test word dataset.

1. Introduction

Recognizing artificial text embedded in images provides high level semantic clues which tremendously enhance automatic image and video indexing. While for printed document, optical character recognition (OCR) systems have already reached high performances, and are widely commercialized, the recognition of superimposed text in images and videos is still a challenging problem.

Kopf et al. [1] apply vertically a shortest-path algorithm to separate the characters in a text line. Then they recognize characters through a matching process based on the curvature scale space (CSS) approach. This approach smoothes the contour of a character with a Gaussian kernel and tracks its inflection points. A recognition rate of 75.6% is reported for a test set of 2986 characters extracted from 20 artificial text images with complex background.

Yokobayashi et al [5, 6] proposed two systems for character recognition in natural scene images. Both of them rely on two steps: the first one is the binarization step and the second one is the recognition step based on an improved version of the global affine transformation (GAT) correlation technique for grayscale character images. The authors report an average recognition rate of 70.3%, ranging from 95.5% for clear images to 24.3% for low contrasted images, from the ICDAR 2003 character sample dataset for the first system and they report an average recognition rate of 81.4%, ranging from 94.5% for clear images to 39.3% for seriously distorted images for the second system.

Previously, in [3], we proposed a character recognition system based on convolutional neural networks that we tested on the same dataset and we reported an average recognition rate of 84.53%, ranging from 93.47% for clear images to 67.86% for seriously distorted images.

As an extension, in this paper, we propose a new optimized text image recognition scheme through what we call the iTRG - image Text Recognition Graph. This is a weighted directed acyclic graph $G = (V, E, W)$, which consists of a set of vertices (or nodes) V, a set of edges (or arcs) E and a set of edge weights W. In the iTRG, the edges are oriented which means that a given edge $e(i, j)$ is oriented from i to j and that the edge $e(j, i)$ does not exist. The edges of the iTRG are also weighted, which means that a weight is assigned to each edge: this weight represents the probability of relevance of its corresponding edge as explained later on. Finally, the iTRG is an acyclic graph which means that there is no cycle nor loop. The proposed scheme consists of building the iTRG, then searching the best path which gives the sequence of characters contained in the processed image.

The remainder of this paper is organized as follows. Section 2 describes in detail the different modules of the proposed scheme, especially the graph design. Experimental results are reported in Section 3. Finally, conclusions are drawn in Section 4.

2 The construction of the iTRG

The construction of the iTRG scheme is based on five main modules:

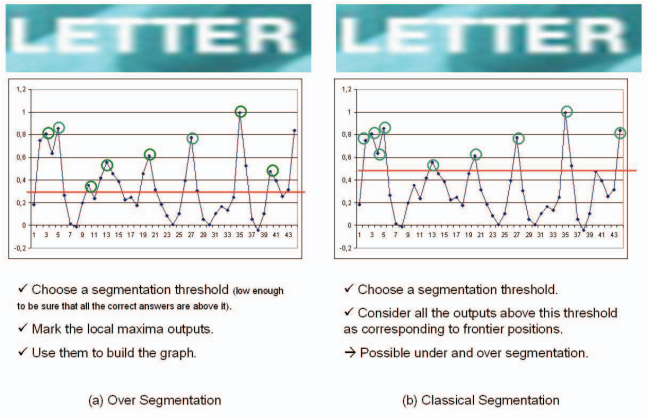

Figure 1. Over segmentation versus classical segmentation

1. The over segmentation module based on the segmentation system presented in [4] with a specific post processing.
2. The graph connection builder module which takes the results of the over segmentation and builds the connections between the graph vertices.
3. The recognition module based on the recognition system presented in [3].
4. The graph weight calculator module based on both results of the over segmentation and the recognition in order to compute the weights of all the edges in the graph.
5. The best path search module that aims to find the path which gives the most probable sequence of characters.

2.1 Over Segmentation

The over segmentation step is based on a method described in [4], relying on a convolutional neural network, that learns how to directly associate to an input text image a vector representing the probability for each column to be a frontier position between two consecutive characters. In order to perform over segmentation, we choose here to apply a lower threshold on the local maxima probabilities as shown in figure 1. Thus, the output of the over segmentation process is a set of several possible frontier positions. The most relevant frontier positions will be determined by the overall process as detailed in the following paragraphs.

2.2 Graph Connection Builder

This module takes the over segmentation module results as input. Thus, every possible frontier position (corresponding to a local maximum output above a chosen threshold) is considered as a vertex in the iTRG. The aim of this module is to build the connections between vertices. Instead of connecting all vertices to each other, we choose to apply some constraints in order to optimize the computation without loosing efficiency. These constraints are:

- A vertex i is connected to a vertex j if the distance between them is within a chosen interval. This means that we have to evaluate the minimum and the maximum width of characters.
- A vertex i cannot be connected to a vertex j if there is a vertex k between them and for which the output is above a chosen threshold. In other words, when a vertex has a high output, we assume that it is a frontier position. Paths which do not include it will not be considered.

2.3 Recognition

The recognition module is based on the character recognition system that we presented in [3]. This module takes as input vertex positions and connections of the iTRG. Each pair of connected vertices corresponds to the borders of a possible character image, which is cropped and processed by the recognition system that outputs the recognized character and its associated probability ($outRec$). The results of this module are given as inputs to the graph weights calculator.

2.4 Graph Weight Calculator

The graph weight calculator module takes as input both results of the over segmentation module and the recognition module to finish the construction of the iTRG. The weights of the edges connecting the vertices are computed as a function of the segmentation and recognition outputs and the distance between to vertices (frontier positions). The equation governing the edge weights is detailed below:

$$edgeweight_{i,j} = outRec_{i,j} \times [Ps(i) \times Ps(j)] \times dist(i,j)$$

- $outRec_{i,j}$ represents the output of the character recognition system applied on the image segment (i,j)
- $Ps(i)$ represents the probability that the position corresponding to the vertex i is a frontier position.
- $dist(i,j)$ is the number of columns separating positions of vertex i and vertex j in the text image.

2.5 Best Path Search

In graph theory, searching the best path is often related to the shortest or longest path problem. In our case, the

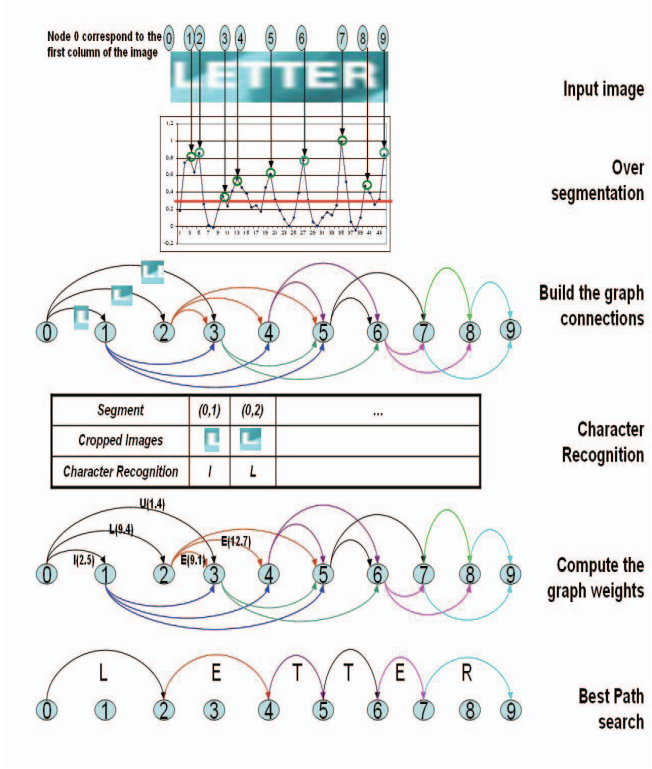

Figure 2. The iTRG

problem consists in finding a path between two vertices such that the sum of the weights of its edges is maximized. We choose to use the Dijkstra algorithm which is a graph search algorithm for a graph with non negative edge weights and that is widely used because of its performance and low computational cost in such problems. Once we have processed the input image with the over segmentation system, built the graph nodes, processed each image segment with the character recognition system, and computed the edge weights, we apply the Dijkstra algorithm to retrieve the best sequence of edges. Labels of this sequence give directly the recognized text. Figure 2 illustrates with an example the whole recognition scheme using the iTRG.

3 Experiments

To test the performance of the iTRG, we have run a series of experiments on a public database in order to contribute to the current state of the art. The ICDAR 2003 conference proposed a competition called robust reading [2] to refer to text images that are beyond the capabilities of current commercial OCR packages. The robust reading competition has been divided into three sub-competitions, namely text locating, character recognition and word recognition. There has been also a competition for the best overall system. Due

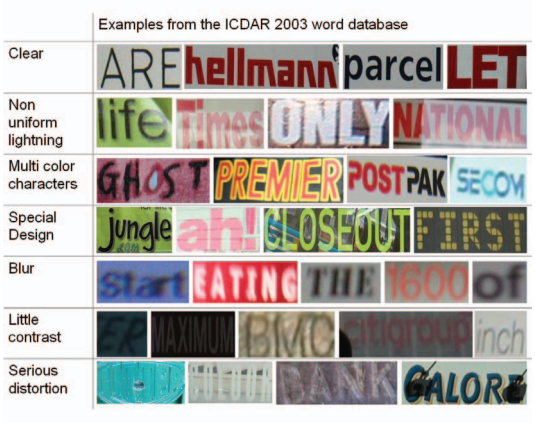

Figure 3. Examples from the ICDAR'03 set

to the complexity of the database, contestants participated only to the text locating competition.

Since the iTRG is conceived to recognize words, we use the ICDAR 2003 test word data set. These word images are of different sizes (26×12, 365×58, 1249×223,...), different fonts, different colors, and present different kinds of distortion. We have manually classified 901 ICDAR word images with a total of 5698 characters into seven groups according to the complexity of the background, the variation of luminance and chrominance and the level of distortion. The seven groups are: clear, non uniform lightning, multi color characters, special design, blur, low contrast, serious distortion. Figure 3 shows some classified examples from the ICDAR 2003 test word dataset. We tested our system according to the performance of the segmentation and the performance of the recognition. For the segmentation, we chose to evaluate the precision and the recall rates. The recall represents the probability that a randomly selected correct border is detected by the system, while the precision represents the probability that a randomly selected detected border is correct.

The ICDAR 2003 public dataset is fully annotated through XML files containing the word written in every image and the character border positions. Given that a border position between two consecutive characters is not unique, we allow a margin of $n = 2$ columns. In other words, if the desired frontier position is P and the system find a border position in the interval $[P - n, P + n]$, then this position is considered as correct. Figures 4 and 5 show the precision and the recall rates of the iTRG when tested on the different categories of the ICDAR 2003 public database. The precision ranges from 63.44% for seriously distorted images to 94.3% for clear images. The recall ranges from 68.97% for seriously distorted images to 92.88% for clear images. These values are compared to the result of the classical scheme consisting of applying the segmentation

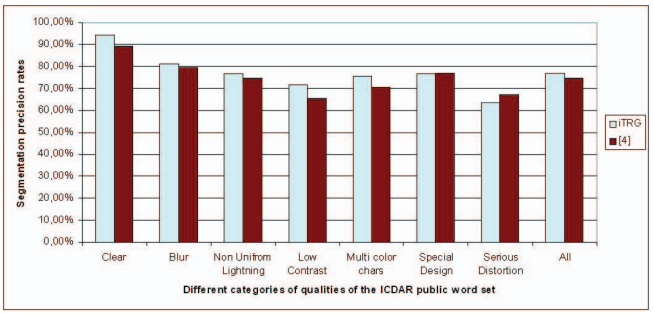

Figure 4. The precision of the segmentation

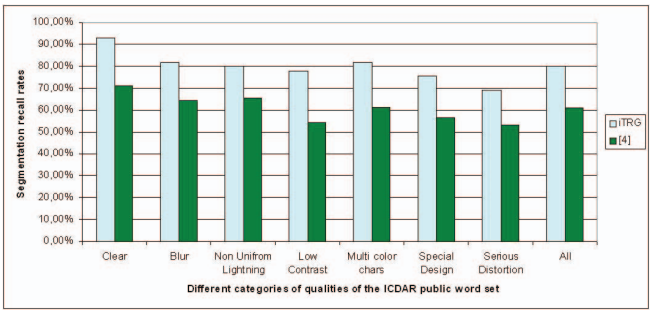

Figure 5. The recall of the segmentation

system then the recognition system, without the use of the iTRG as detailed in [4].

We remark that there is an overall enhancement of 2% in precision and 19% in recall. In fact, the over segmentation increases the probability to get the correct border positions among the set of detected borders which increases the recall and the recognition outputs that helps in choosing the best borders increases the precision. This is especially true for images with low contrast where the enhancement reaches 6.32% in precision and 23.6% in recall and for images with multi-color characters where the enhancement reaches 4.66% in precision and 20.46% in recall.

Figure 6 shows the word recognition rates of the iTRG on the different categories of the ICDAR 2003 public set

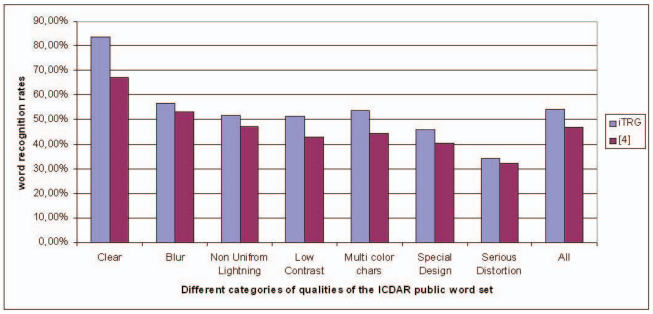

Figure 6. Recognition rates of the iTRG

compared to those of the classical scheme [4]. The iTRG recognition performance is by far better than the classical scheme performance. There is an enhancement of 10% over the whole data set, an enhancement of 10% for images with low contrast and with multi-color characters and an enhancement of more than 15% for clear images.

4. Conclusion

In this paper, we have presented a robust text recognition system based on a specific directed acyclic graph. To build this graph, we applied text segmentation and character recognition modules based on convolutional neural networks. We contributed to the state-of-the-art comparison efforts initiated by ICDAR 2003 by evaluating the performance of our method on the public ICDAR 2003 test word set. The results reported here are encouraging. However, there are still errors due to the confusion between similar characters such as 'l' and 'I'. In order to overcome these limitations, we are working towards including a statistical language modeling module in our scheme.

References

[1] S. Kopf, T. Haenselmann, and W. Effelsberg. Robust character recognition in low-resolution images and videos. Technical report, Department for Mathematics and Computer Science, University of Mannheim, Apr. 2005.

[2] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamaoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J. Jolion, L. Todoran, M. Worring, and X. Lin. Icdar 2003 robust reading competitions: Entries, results and future directions. *International Journal on Document Analysis and Recognition*, 7(2–3):105–122, June 2005.

[3] Z. Saidane and C. Garcia. Automatic scene text recognition using a convolutional neural network. In *Proceedings of the Second International Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*, Sept. 2007.

[4] Z. Saidane and C. Garcia. An automatic method for video character segmentation. In *Proceedings of the Fifth International Conference on Image Analysis and Recognition (ICIAR)*, volume 2, pages 874–879, June 2008.

[5] M. Yokobayashi and T. Wakahara. Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation. *Eighth International Conference on Document Analysis and Recognition (ICDAR)*, 1:167–171, Aug. 2005.

[6] M. Yokobayashi and T. Wakahara. Binarization and recognition of degraded characters using a maximum separability axis in color space and gat correlation. *18th International Conference on Pattern Recognition (ICPR)*, 2:885–888, Aug. 2006.