An Automatic Method for Video Character Segmentation

Conference Paper · June 2008

DOI: 10.1007/978-3-540-69812-8_55 · Source: DBLP

2 authors, including:

Christophe Garcia
Institut National des Sciences Appliquées de Lyon
164 PUBLICATIONS **3,336** CITATIONS

An Automatic Method for Video Character Segmentation

Zohra Saidane and Christophe Garcia

Orange Labs,
4 rue Clos Courtel, 35510 Cesson Sévigné, France
{zohra.saidane,christophe.garcia}@orange-ftgroup.com

Abstract. This paper presents an automatic segmentation system for characters in text color images cropped from natural images or videos based on a new neuronal architecture insuring fast processing and robustness against noise, variations in illumination, complex background and low resolution. An off-line training phase on a set of synthetic text color images, where the exact character positions are known, allows adjusting the neural parameters and thus building an optimal non linear filter which extracts the best features in order to robustly detect the border positions between characters. The proposed method is tested on a set of synthetic text images to precisely evaluate its performance according to noise, and on a set of complex text images collected from video frames and web pages to evaluate its performance on real images. The results are encouraging with a good segmentation rate of 89.12% and a recognition rate of 81.94% on a set of difficult text images collected from video frames and from web pages.

Key words: Video OCR, Text Segmentation, Character recognition, Neural Networks, Image Processing

1 Introduction

The tremendous increase in the use of multimedia not only as a communication medium, but also as an educational and entertaining medium, entails a need to powerful indexing systems, so as to ensure easy access to the relevant information, navigation, and organization of vast repositories of multimedia (audio, images, video).

Multimedia entities carry two types of information, content based information and concept based information. The first kind consists of low level features (colors, shapes, edges, frequencies, energy of the signal), and could be automatically identified and extracted, whereas the second consists of high level features, called also semantic features, which indicates what is represented in the image, or what is meant by the audio signal, and are in general hardly identified automatically.
One of the goals of the multimedia indexing community is to design a system able

of producing a rich semantic description of any multimedia document. Among the semantic features, the text embedded in images is of particular interest. Firstly it is very useful for describing the contents of an image and secondly it enables applications such as keyword-based image search, automatic video logging, and text-based image indexing.

Character recognition systems usually require a preprocessing step, called segmentation, which split the text image into isolated character images. This task is challenging because on one hand text images have usually complex background, noise, illumination variation and on the other hand a part of a character image may be confusing with another character. For example, a part of the letter 'm' could be 'n', and a part of the letter 'w' could be 'v' which is very confusing.

In the state of the art of automatic character segmentation, there are three main classes of approaches, the binarization based, the dissection based and the recognition based segmentation. The binarization based methods, which are the most popular, [1],[2],[3],[4],[5],[6] consist in converting the text pixels to black and the background pixels to white. Most of them rely on global or local discriminating thresholds determined according to some statistical analysis of the luminance or chrominance distribution generally based on histograms. They are usually combined with classical OCRs which provide good results on printed documents.

The dissection methods are based on the decomposition of the image into a sequence of regions using general features. For example, the projection profile analysis used by [7] consists of a simple running count of the black pixels in each column so that it can detect the white space between successive letters. This technique has to be applied on binarized image. Another dissection technique is the split and merge algorithm, proposed Horowitz and Pavlidis [8] and used by [9] for text web images segmentation based on the human colour perception properties. Karatzas and Antonacopoulos [9] use two homogeneity criterions for the split and merge algorithm, the first one is based on the lightness histogram and the second one based on the hue histogram. For each histogram, peaks are identified and then analysed. The splitting and merging process are then conducted depending on whether adjacent peaks are perceived to be similar by a human observer or not. In the merging process, connected component analysis is also used. This technique reach a rate of 69.65% of well identified characters, when tested on a variety of representative web images. Kopf et al. [10] propose a three-step dissection approach. First, the resolution is enhanced by linear interpolation to scale the text region by a factor of four. Then separators for individual characters are located. For this purpose, they assume that the contrast between text pixels and background pixels is high, whereas the average difference between adjacent background pixels is much lower. Therefore they search a path from the top to the bottom of the text region based on the Dijkstra shortest-path algorithm for graphs. The cost of a path is defined as the summarized pixel differences between neighbor pixels (left, right, down). The path with the minimum cost, called cheapest path, rarely crosses character pixels. It

defines a good separator for characters. They report an error rate of 9.2% when tested on twenty images with complex backgrounds.

The recognition based segmentation methods [11],[12],[13] are also known as segmentation free methods, because no complex cutting process is used but rather a decision on the character region based on the recognition result. Therefore, the recognition system is the most important module in these techniques.

In this paper, we propose an automatic segmentation system for characters in text color images cropped from natural images or videos based on a new neuronal architecture insuring fast processing and robustness against noise, variations in illumination, complex background and low resolution.

The remainder of this paper is organized as follows. Section 2 describes in detail the architecture of the proposed neural network and the training process. It explains also how to use this system combined with a character recognition system. Experimental results are reported in Section 3. Conclusions are drawn in Section 4.

2 The Segmentation System

2.1 Architecture of The Network

Our character segmentation approach relies on a novel architecture of convolutional neural networks, which are known for their very good performance to solve various pattern recognition problems [14],[15],[6]. As shown in Figure 1, the architecture of the proposed neural network consists of four layers. Each layer contains a set of feature maps which are the results of convolutions. Applying and combining these automatically learnt operations insure the extraction of robust features, leading to the automatic detection of border positions between consecutive characters in a text image.

The first layer is the input layer E; it consists of $N_E = 3$ input maps, each of them corresponding to one color channel of the image, depending on the color space (RGB, YUV, CMY, etc.). Their pixel values are normalized to the range [-1, 1]. The RGB color space has been chosen in our experiments.

The second layer is a convolution layer C_1 which is composed of NC_1 maps. Each unit in each map is connected to a $M_1 \times 1$ neighborhood in each map of the previous layer. Furthermore, the trainable weights forming the receptive field, which play the role of convolutional mask, are forced to be equal for the entire map. This idea of weight sharing insures the extraction of the same feature over the entire map and reduces the computational cost. A trainable bias is added to the results of each convolutional mask. Each map can be considered as a feature map that has a learnt fixed feature extractor that corresponds to a pure convolution with a trainable mask, applied over the maps in the previous layer. Multiple maps lead to the extraction of multiple features.

The third layer is also a convolution layer C_2. It is composed of NC_2 maps. Each unit in each map is connected to a $M_2 \times 1$ neighborhood in the maps of the previous layer. Here again, we apply the idea of weight sharing. Layer C_2 allows

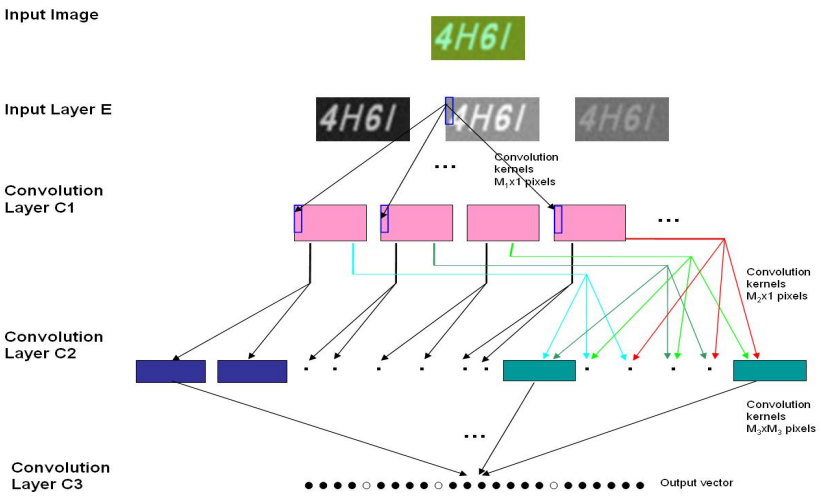

Fig. 1. Architecture of the segmentation neural network

extracting more complex information; outputs of different feature maps of layer C_1 are fused in order to combine different features. The combination scheme is explained in table 1, it shows two kinds of feature maps, the first category is generated from only one previous feature map and it contains $2 \times NC_1$ and the second category is generated from the combination of two previous feature maps and it contains $\frac{NC_1 \times (NC_1 - 1)}{2}$.

Compared to the classical convolutional neural network architecture, the proposed architecture does not contain sub-sampling layers, which are generally used for ensuring tolerance versus pattern . Our goal being to detect the border position between consecutive characters, the exact positions of the extracted features are important.

The last layer, as the previous two layers, is a convolution layer C_3. However, there is only one map in this layer and the convolutional mask is not a vector, but rather a matrix. Indeed, each unit the map of layer C_3 is connected to a $M_3 \times M_3$ neighborhood in the maps of the previous layer. Moreover, M_1, M_2 and M_3 are chosen so that the height of the map of the last layer C_3 is equal to 1. Thus, this map is a vector, where the units with high outputs correspond to a border position between two consecutive characters in the input image.

2.2 Implementation

Data To train such a network, we need a set of color text images where the border positions between every two consecutive characters are known. To avoid

the costly manual text segmentation, and because we are convinced of the good generalization ability of the proposed neural network, we choose to use synthetic images that we build automatically. Thus, a set of color text images of size $W \times H = 48 \times 23$ is constructed using different fonts, different text colors, different background colors, and different types of noise. Figure 2 shows some examples of training images.

Fig. 2. Examples of images of the training set

Our training set contains 5000 RGB color text images, $N_t = 4500$ for training and $N_v = 500$ for validation. For each of them, we build the desired output vector D_h. It is a vector of width N_o elements equal to the image width W minus $(M_3 - 1)$ because of the convolution border effects. An element of this vector is set to 1 if its position correspond to a border position between two consecutive characters, and is set to -1 otherwise.

Training Phase We choose to build $NC_1 = 6$ maps in layer C_1, and $NC_2 = 27$ maps in layer C_2. Layer C_1 contains only one map. We use sigmoid activation functions for the three convolution layers C_1, C_2, and C_3. The convolution window sizes are $M_1 = 13$, $M_2 = 7$ and $M_3 = 5$. The combination scheme between layer C_1 and layer C_2 is explained in the table 1:

The different parameters governing the proposed architecture, i.e., the number of layers, the number of maps, as well as the size of the receptive fields, have been experimentally chosen.

The training phase was performed using the classical back-propagation algorithm with momentum modified for being used in convolutional networks as described in [14]. At each iteration, the three RGB channels of the color text image are presented to the network as inputs and the border positions between consecutive characters as the desired output vector. The weights are updated as a function of the error between the obtained output vector and the desired vector.

Table 1. Combination Scheme between layer C1 and layer C2.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0	X	X										X	X	X	X	X											
1			X	X								X					X	X	X	X							
2					X	X							X				X				X	X	X				
3							X	X						X				X			X			X	X		
4									X	X					X				X			X		X		X	
5											X	X				X				X			X		X	X	

The objective of the network being to obtain final values $\{F_{h,k}\}_{h=1..N_o,k=1..N_v}$ equal to the desired values $\{D_{h,k}\}_{h=1..N_o,k=1..N_v}$, we classically choose to minimize the MSE (Mean Square Error) between obtained and desired values over a validation set of $N_v = 500$ images.

$$MSE = \frac{1}{N_v \times N_o} \sum_{k=1}^{N_v} \sum_{h=1}^{N_o} (F_{h,k} - D_{h,k})^2 \tag{1}$$

Testing Phase Once the training is achieved, the system is ready to segment color text images a follows:

1. Resize the text image to the retina height, while preserving the aspect ratio (The size of the training input images are called retina size).
2. Feed the network with the resized text image as input
3. Retrieve the output vector of the network and save the column indexes corresponding to the border positions detected
4. Crop the text image according to these column indexes
5. Each cropped symbol is considered as a character image.

We have to mention here that the width of the image must not be equal to the retina width, because of the sharing weights property of this network. In fact, for each of the three layers , there is only one convolution mask per map which corresponds to the matrix of weights. This convolution mask can be applied as much as necessary according to the width of the image processed.

Once character images are available, a character recognition system can be applied. We choose to use a character recognition system based also on convolutional neural network, that we proposed recently [16] and which yields very good results compared to the state of the art systems.

3 Experimental Results

To test the performance of our method, we use two databases. The first one contains 5000 synthetic text color images, where the exact positions of the different characters are known. This database is used to evaluate the performance of our segmentation system according to noise. This database includes five categories

of 1000 images with a gaussian noise variance ranging from 3 to 11 by a step of 2. For this evaluation, it is easy to compute the recall and the precision criteria since the character positions are known. We remind here the formula of Recall and Precision.

$$Recall = \frac{NumberofCorrectlySegmentedCharacters}{NumberofDesiredCharacters} \tag{2}$$

$$Precision = \frac{NumberofCorrectlySegmentedCharacters}{NumberofObtainedCharacters} \tag{3}$$

Given that a border position between two consecutive characters is not unique, we allow a margin of n columns. In other words, if the desired fontier position is P and the system find a border position beetween $P-n$ and $P+n$, then this position is considered as correct. A margin of $n = 2$ seems to be acceptable.

Figure 3 shows the performance of our segmentation system as a function of the gaussian variance. The variation of recall and precision does not exceed 2% and remains always above the 80%.

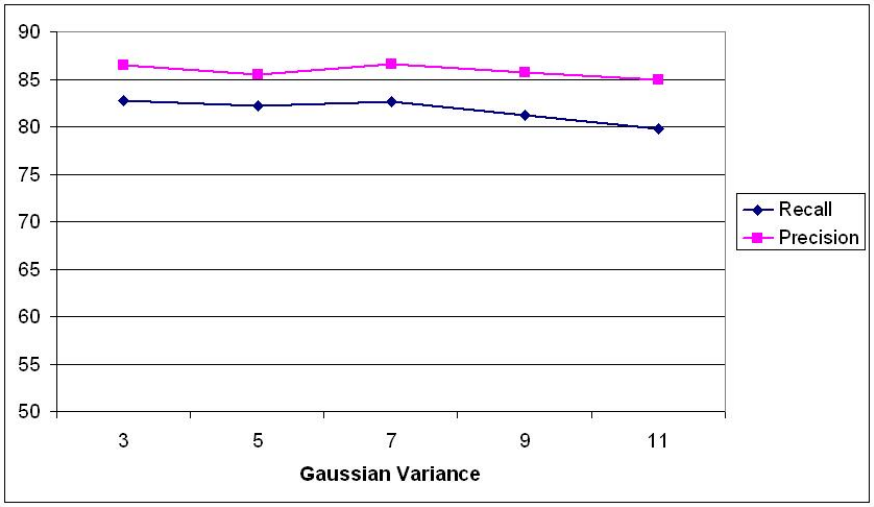

Fig. 3. Performance of the segmentation system as function of noise

The second test database contains 52 real images collected from video frames and web pages. Figure 4 shows some examples from the test images data set. We apply the proposed segmentation system and we manually evaluate the number of corrected segments. In the test set, we have 469 characters. We have found 418 correctly segmented characters, which means that our algorithm reach a correct segmentation rate of 89.12%.

We compare our segmentation method to the following four state of the art techniques:

Fig. 4. Some segmentation test examples

1. the Niblack method [1] computes a threshold based on the mean and the standard deviation of each block in the image, then binarize the image.
2. the Sauvola method [2] is also based on the mean and the standard deviation to binarize the text image.
3. the Lienhart method [5] consists in choosing the intensity separating value halfway between the intensity of the text color and the background color as a binarization threshold.
4. the CTB method [6] is a recent binarization method based on convolutional neural network.

A classical way of comparing the segmentation methods performance is to combine them with an OCR and to compare the recognition rates obtained. For the state of the art methods, two OCRs (Tesseract a public software and Abby FineReader 8.0 a commercial software) are used to get recognition results. And for our segmentation system, we use the CNNOCR [16] since it is does not need a binarized image as the classical OCR systems do. Figure 5 shows the corresponding recognition rates.

While our method outperforms the classical techniques [1],[2],[5], it does not outperform the CTB combined with the commercial OCR, FineReader. Nevertheless, we think that the results are encouraging and improvement could be achieved on the overall scheme by considering some linguistic analysis as most of the classical OCR systems. This will be considered in a future work.

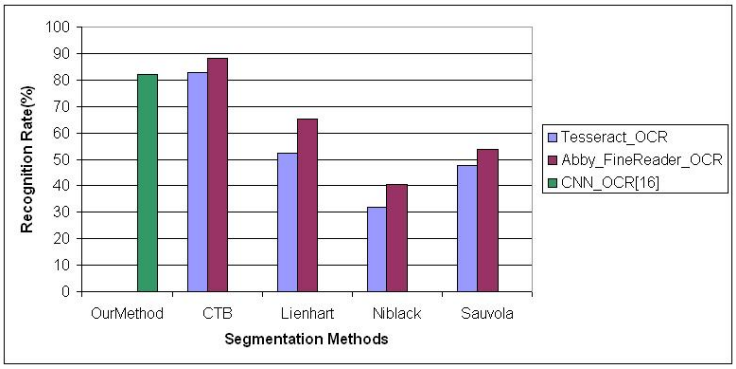

Fig. 5. Comparison of our method with other state of the art techniques

4 Conclusion

In this paper, we have proposed a novel approach based on a specific convolution neural network architecture designed for complex color text image segmentation. This approach is a one step approach which is robust to noise, low resolution and complex background. Evaluated on a set of text images collected from video frames and web pages, our method gives encouraging results. We believe that the combination of our segmentation system with a character recognition system proposed by [16] is very promising, and we plan as a future work to improve the overall scheme by applying some linguistic models.

References

1. Niblack, W.: An Introduction to Digital Image Processing. N.J.: Prentice Hall (1986)
2. Sauvola, J., Seppnen, T., Haapakoski, S., Pietikinen, M.: Adaptive document binarization. In International Conference on Document Analysis and Recognition **1** (1997)
3. Liao, P.S., Chen, T.S., Chung, P.C.: A fast algorithm for multilevel thresholding. Journal of information science and engineering (2001)
4. Garcia, C., Apostolidis, X.: Text detection and segmentation in complex color images. IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP'00 **4**(2326–2329) (June 2000)
5. Lienhart, R., Wernicke, A.: Localizing and segmenting text in images and videos. IEEE Transactions on circuits and systems for video technology **12**(4) (April 2002)
6. Saidane, Z., Garcia, C.: Robust binarization for video text recognition. In: Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR). Volume 2. (September 2007)
7. Sato, T., Kanade, T., Hughes, E., Smith, M.: Video ocr for digital news archive. In: Proceedings of the International Workshop on Content-Based Access of Image and Video Databases (CAIVD '98). (January 1998)
8. Horowitz, S., Pavlidis, T.: Picture segmentation by a tree traversal algorithm. ACM **2** (1976)

9. Karatzas, D., Antonacopoulos, A.: Text extraction from web images based on a split-and-merge segmentation method using colour perception. In: Proceedings of the 17th International Conference on Pattern Recognition. ICPR 2004. Volume 2. (August 2004)
10. Kopf, S., Haenselmann, T., Effelsberg, W.: Robust character recognition in low-resolution images and videos. Technical report, Department for Mathematics and Computer Science, University of Mannheim (April 2005)
11. Burges, C., Matan, O., LeCun, Y., Denker, J., Jackel, L., Stenard, C., Nohl, C., Ben, J.: Shortest path segmentation: A method for training a neural network to recognize character strings. In: Proceedings of the International Joint Conf. Neural Networks. Volume 3. (1992)
12. Chen, D., Odobez, J., Thiran, J.: Monte carlo text segmentation. Int. journal of Pattern Recognition and Artificial Intelligence (2005)
13. Sato, T., Kanade, T., Hughes, E., Smith, M., Satoh, S.: Video ocr: Indexing digital news libraries by recognition of superimposed captions. In: ACM Multimedia Systems Special Issue on Video Libraries. (February 1998)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient based learning applied to document recognition. Proc. of the IEEE (November 1998)
15. Garcia, C., Delakis, M.: Convolutional face finder: A neural architecture for fast and robust face detection. IEEE Transactions on pattern analysis and machine intelligence **26**(11) (November 2004)
16. Saidane, Z., Garcia, C.: Automatic scene text recognition using a convolutional neural network. In: Proceedings of the Second International Workshop on Camera-Based Document Analysis and Recognition (CBDAR). (September 2007)