HAL
open science

Text Recognition in Multimedia Documents: A Study of two Neural-based OCRs Using and Avoiding Character Segmentation

Khaoula Elagouni, Christophe Garcia, Franck Mamalet, Pascale Sébillot

To cite this version:

Khaoula Elagouni, Christophe Garcia, Franck Mamalet, Pascale Sébillot. Text Recognition in Multimedia Documents: A Study of two Neural-based OCRs Using and Avoiding Character Segmentation. International Journal on Document Analysis and Recognition, 2014, 17 (1), pp.19-31. 10.1007/s10032-013-0202-7 . hal-00867225

HAL Id: hal-00867225

https://hal.science/hal-00867225v1

Submitted on 27 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IJDAR manuscript No.
(will be inserted by the editor)

Text recognition in multimedia documents: A study of two neural-based OCRs using and avoiding character segmentation

Khaoula ELAGOUNI · Christophe GARCIA · Franck MAMALET · Pascale SÉBILLOT

Received: date / Accepted: date

Abstract Text embedded in multimedia documents represents an important semantic information that helps to automatically access the content. This paper proposes two neural-based OCRs that handle the text recognition problem in different ways. The first approach segments a text image into individual characters before recognizing them, while the second one avoids the segmentation step by integrating a multi-scale scanning scheme that allows to jointly localize and recognize characters at each position and scale. Some linguistic knowledge is also incorporated into the proposed schemes to remove errors due to recognition confusions. Both OCR systems are applied to caption texts embedded in videos and in natural scene images and provide outstanding results showing that the proposed approaches outperform the state-of-the-art methods.

Keywords OCR · Character segmentation · Convolutional neural network · Language model

1 Introduction

Textual patterns embedded or captured in images and videos provide high-level semantic clues often interesting for applications and services such as multimedia document indexing and retrieval, teaching videos and robotic vision systems. In these contexts, the design of efficient OCR (Optical Character Recognition) systems specifically adapted to multimedia documents is an important issue. However, the huge diversity of texts, especially in natural scene images, and the difficult acquisition conditions (low resolution, complex background, non uniform lighting, occlusions and blurring effects) make the task of text recognition a challenging problem that has raised a growing interest in recent research activities [3,34,42,11].

In this paper, we propose to study and compare two different approaches for text recognition in images. The first one [11] consists in segmenting the text image into individual characters before recognizing them. In contrast with existing methods, this OCR system performs a non-linear character segmentation taking into account the local morphology of text images, in order to improve character recognition performance. The second OCR system [10] avoids explicitly segmenting characters and addresses the problem in a way different from prior approaches using a multi-scale scanning process that allows to recognize characters at their appropriate scale and position within the whole text image. In both methods, in order to tackle the high variability of the input images, we propose to rely on ConvNets (Convolutional Neural Networks [19]) that are particularly a robust pattern recognition technique. Another contribution of our work consists in introducing, for both methods, a supervision scheme based on language models in order to remove some errors due to recognition confusions.

In the community of text recognition, texts in multimedia documents have been classified into two categories [16]: "caption texts" (*cf.* Fig. 1.C), which are ar-

K. ELAGOUNI
Orange Labs R&D, 35512 Cesson-Sévigné, France.
E-mail: khaoula.elagouni@orange.com

C. GARCIA
LIRIS / INSA de Lyon, 69621 Villeurbanne, France.
E-mail: christophe.garcia@liris.cnrs.fr

F. MAMALET
Orange Labs R&D, 35512 Cesson-Sévigné, France.
E-mail: franck.mamalet@orange.com

P. SÉBILLOT
IRISA / INSA de Rennes, 35042 Rennes, France.
E-mail: pascale.sebillot@irisa.fr

(A) (B) (C)

Fig. 1 Examples of texts in multimedia documents: (A) and (B) are "scene texts" and (C) is a "caption text".

tificially overlaid on images or videos, and "scene texts" (*cf.* Fig. 1.A and 1.B), which exist naturally in images or videos. Here, in order to evaluate and compare the proposed OCR methods, which is the focus of this paper, new experiments are carried out on two difficult datasets: one containing "caption texts" embedded in digital videos and the other containing natural "scene text" images. Performed experiments show that both designed OCRs obtain outstanding results and outperform the other state-of-the-art methods. The results are analyzed and discussed, highlighting the benefits and limits of both approaches.

The remainder of this paper is organized as follows. After a review of state-of-the-art methods dedicated to text recognition techniques in multimedia documents (section 2), the two proposed OCRs are detailed in sections 3 and 4. The integration of a language model is described in section 5. The evaluation of the proposed methods as well as comparisons with state-of-the-art approaches are presented and discussed in section 6. Finally, conclusions are drawn in section 7.

2 Related work

Since years, OCR systems have been an important application of pattern recognition and computer vision. In these research domains, prior works have mainly focused on systems operating on scanned documents and on handwritten texts. Recently, a considerable progress has been made in the specific field of text recognition in images and videos. A review of the new advances achieved in text recognition in multimedia documents is presented in [36]. Different issues related to this recognition problem have been identified, including text detection [23,21,46,6], text enhancement and binarization [15,3,47,48,24] (a pre-processing step that aims to improve recognition performance), character segmentation [2,26,34,31], character recognition [4,17,32] and the integration of linguistic knowledge [49,45,11].

In this paper, we do not focus on the detection of text, but on the steps involved in the text recognition task, *i.e.*, text image pre-processing, character segmentation, character recognition and text recognition. Related works are presented in this section.

2.1 Text image pre-processing

Most OCR methods rely on pre-processing treatments, and specifically on binarization in order to ease the recognition step. Saïdane *et al.* [33] introduced an automatic binarization step based on a ConvNet particularly robust to complex background and low resolution. The main idea is to automatically learn the parameters of the ConvNet to transform a text image into a binarized version, using a large training set. Mishra *et al.* [28] presented a Markov Random Field (MRF) based technique of binarization adapted to scene text images. Li *et al.* also proposed a method dedicated to scene text images where local visual information and contextual label information are integrated in a Conditional Random Field (CRF) [22]. Another binarization approach using text contours and a local thresholding method was proposed by Zhou *et al.* [50]. Recently, Wakahara *et al.* [41] defined a binarization method that relies on a K-means clustering and a Support Vector Machine (SVM) model to binarize color scene text images with complex background, while Ntirogiannis [30] used the stroke width and convex hull analysis to binarize texts embedded in videos. For video data, Hua *et al.* [15] and Yi *et al.* [47] were interested in solving problems related to complex backgrounds by using multiple frame integration. The main idea consisted in taking advantage of the temporal redundancy of a text appearing in successive frames of a video.

2.2 Character segmentation

Among state-of-the-art methods, two major approaches can be distinguished: *segmentation-based* approaches which segment the text into individual characters before the recognition step, and *segmentation-free* approaches which recognize a succession of characters directly from the whole text image without any segmentation.

Casey *et al.* provide a complete survey of character segmentation methods in [2]. In [27] and [35], segmentation methods that rely on a classical projection profile technique are presented and applied to caption texts extracted from digital videos. Shivakumara *et al.* propose a gradient-based character segmentation scheme [37], while Phan *et al.* use a gradient vector flow-based method to segment characters [31]. To improve performance, other authors [26,34] propose hybrid approaches that combine image processing techniques and recognition results. The key point is to build concurrent segmentations and to rely on a character recognizer to identify the correct ones.

In contrast to these methods, other authors propose OCR systems that do not rely on conventional segmen-

tation techniques. Kusachi *et al.* [18] use a coarse-to-fine scanning technique and classify clipped regions to recognize characters. Wang *et al.* [42] present a word recognition approach that relies on a generic object recognition method, in which words are considered as object categories. Words are considered as sequences of characters identified and localized in text images based on some extracted features. However, the major issue of these approaches is the difficulty to choose the discriminant features to represent extremely variable characters.

2.3 Character recognition

Other works have focused on the problem of single character recognition in images and videos. Among these single character recognizers, two main approaches can be distinguished: pattern matching methods and machine learning methods.

In the first category, characters are usually identified by a set of features. First, a database of models of features is generated. Then, for each image corresponding to a character, features are extracted and matched against the database in order to recognize the character class. In [17], edges and contours are considered as features characterizing characters, while in [4], for each binarized image character, four side-profiles are extracted and matched to recognize characters. Side-profiles are obtained by counting the white pixels in each direction (left, right, up and down) until encountering black pixels. Halima *et al.* also used a projection profile technique to recognize Arabic character images extracted from digital videos [13]. Negishi *et al.* proposed instead to use corners and curves that are matched relying on a voting algorithm [29]. Recently, inspired by speech recognition, Som *et al.* [39] designed an OCR system that uses an Hidden Markov Model (HMM) to identify characters as a sequence of states. However, as in any pattern recognition problem, the major issue is to define the robust features that represent characters independently of the image resolution and the background complexity. Therefore, performance of these methods may be very variable depending on the chosen features and the image conditions.

In the second category, methods are designed to learn automatically how to classify characters either directly from their images or after extracting features. In [32], Saïdane and Garcia have presented an automatic method for scene character recognition based on a convolutional neural classification approach. The system is able to deal directly with the raw pixels of extremely variable characters and appears to be particularly robust to different image distortions. Another work presented in [8] relies on a SVM classifier which learns how to recognize characters from image pixels and which also obtains good results. Aiming at recognizing Kanji characters captured in natural scene images, a voting method was chosen by Kusachi *et al.* [18] to identify characters with recognition dictionaries obtained by patterns learning. Recently, a method based on unsupervised features learning was proposed to detect and recognize characters in natural scene images [5].

2.4 Text recognition

Methods that rely on a character segmentation based recognition often use a graph able to handle different concurrent segmentations [34]. Free-segmentation approaches propose a peak detection system to recognize texts [18]. The idea is to analyze extracted features (segments of character candidates) in order to detect peak points present in these features and eliminate the others. Texts are thus recognized as the sequences of characters identified by means of the peak detection. Further information, such as language properties and lexicons, can also be integrated to improve the performance of OCR systems [43].

In this paper, we present two approaches for text recognition where no binarization step is required: one relies on a segmentation step well-adapted to the local morphology of images while the other uses a multi-scale scanning scheme which avoids character segmentation. We also propose a robust character recognizer based on a neural model able to learn how to extract relevant features and identify characters without any preprocessing step. Some linguistic knowledge is integrated in both OCRs to avoid the drawbacks of local character-by-character recognition and improve performance.

3 The segmentation-based OCR

This section presents our first OCR approach [11] designed to recognize texts captured in natural scene images and embedded in videos. Figure 2 depicts the outline of this approach whose first step consists in a character segmentation.

3.1 Character segmentation

In order to find reliable separations between characters, we start by analyzing the text image to distinguish the text from the background. Assuming that pixels of a text image are of two classes, “text” and “background”,

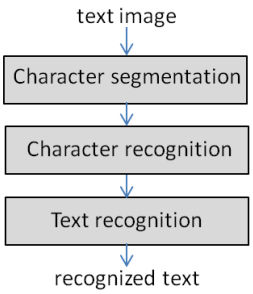

Fig. 2 Outline of the segmentation-based OCR.

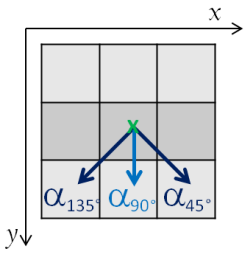

Fig. 3 The shortest path computation: the three allowed directions and their respective weights.

and that their respective intensities are governed by Gaussian distributions, a Gaussian mixture is fit to the image intensity histogram. The estimation of both distributions can thus be performed by maximizing the likelihood between a set of observations—the image intensity histogram—and a Gaussian mixture model. Using an Expectation-Maximization (EM) algorithm [7], parameters of both distributions are obtained, and then used to generate a fuzzy map indicating, for every pixel, its membership degree to the class "text". To do so, a model is applied such as the membership value is 0 if $i \leq \mu_1$, 1 if $i \geq \mu_2$ and varies linearly between these bounds, with i being the intensity in the fuzzy map and μ_1 and μ_2 the means of the distributions.

In the case of video data, the temporal redundancy of each text is also taken into consideration to generate another fuzzy map. Since intensity distributions and temporal variation represent two independent sources of information, the two fuzzy maps are then combined to obtain a more accurate one. To do so, a fusion system, with an adaptive behavior depending on the values to combine, is required. According to [44], the chosen operator should be conjunctive (with a severe behavior) if both values are low, disjunctive (with an indulgent behavior) if both values are high and depends only on the intensity distribution analysis if the temporal variation is low. The operator expressed by eq. 1 satisfies these conditions:

$$f(x,y) = \begin{cases} x & if\ y \leq th \\ \sigma(x,y) = \frac{g(x,y)}{g(x,y)+g(1-x,1-y)} & otherwise \end{cases} \quad (1)$$

where x refers to the intensity analysis result, y refers to the temporal variation result and th is a threshold determined empirically. $\sigma(x,y)$ is the associative symmetric sum, and $g(x,y)$ is a positive increasing function.

In contrast with other state-of-the-art methods that search for linear segmentations to separate characters, we propose to segment characters by non-linear borders which are well-suited to different morphologies. This is done by using the obtained fuzzy map in order to enhance recognition performances. Each segmentation border is computed as the shortest vertical path containing pixels of low probabilities (*i.e.*, membership degrees) to belong to the class "text". Considering the fuzzy map as a grid of vertices and using the shortest path algorithm, segmentations are determined as paths connecting pixels from the top to the bottom of the image without crossing any pixel of class "text" (typically, pixels having a membership degree over a fixed threshold). Three directions are allowed in our algorithm: 45°, 90°, and 135° with respect to the horizontal axis. Three weights $\alpha_{45°}$, $\alpha_{90°}$ and $\alpha_{135°}$, fixed empirically, are respectively assigned to each direction, with $\alpha_{45°} = \alpha_{135°}$ (*cf.* Fig. 3). Typically, assuming that each pixel in the fuzzy map is identified by its coordinates (x,y) and characterized by its intensity I, the shortest path $Path(S)$ which starts from one pixel S of the first line of the image is computed with the following formula:

$$Path(S) = \left\{A_i, A_{i+1} = \underset{B}{\operatorname{argmax}} \left(\alpha_{(A_i,B)} \cdot I_B\right)\right\}_{i \in \{0,\ldots,n\}} \quad (2)$$

where $A_0 = S$ is the first pixel in the path, B is a pixel of the image that satisfies the two conditions $|x_B - x_{A_i}| < 2$ and $y_B = y_{A_i} + 1$, and $\alpha_{(A_i,B)}$ is equal to $\alpha_{90°}$ if $x_B = x_{A_i}$ and to $\alpha_{45°}$ otherwise.

The resulting segmentation borders are characterized by the value of their highest pixel probability. In the rest of the paper, this value is called the score of the path. Two categories of segmentation borders are distinguished depending on their cost: "accurate" ones with low costs (under a threshold set empirically) and "risky" ones with higher costs. "Accurate" paths are considered as corresponding to correct separations between two characters while "risky" ones will be questioned later relying on further information: character recognition results (*cf.* subsection 3.3) and linguistic knowledge (*cf.* subsection 5). Figure 4 illustrates an example of the obtained non-linear segmentations.

According to the survey of character segmentation techniques presented by Casey *et al.* [2], our method can be considered as a hybrid method which takes advantages of both of "dissection" and "recognition-based" techniques. Indeed, "accurate" segmentations are obtained by an intelligent process including an analysis

Fig. 4 An example of non-linear segmentations: "accurate" ones are shown in green and "risky" ones are shown in red.

of the image and without any symbol classification. Thus, they can be considered as deriving from "dissection techniques". In contrast, "risky" segmentations that will be discussed in accordance with recognition results can be considered as derived from "recognition-based techniques".

3.2 Character recognition

Once segmented, characters have to be recognized. Among state-of-the-art approaches, the dominant methodology consists first in binarizing the images and then extracting visual features to recognize characters. The main drawback of this kind of methods is that binarization may fail when the background is complex, leading to poor recognition rates. Unlike these techniques, we propose to train a neural network able to learn to recognize characters directly from the input image. Convolutional Neural Networks (ConvNets) are bio-inspired hierarchical multi-layered neural networks proposed by LeCun *et al.* [19] to learn visual patterns directly from the image pixels without any pre-processing step. Relying on specific properties (namely local receptive fields, weight sharing and sub-sampling), this neural model is particularly robust to noise, geometric transformations and distortions and has shown a great ability to deal with a large number of extremely variable patterns. This neural model has been used in many classification tasks [38] ranging from handwritten character recognition [20] to face detection [12] and, it generally outperforms other classification models such as SVMs [11].

For our character recognition problem, several configurations were tested on our datasets. The ConvNet, hereafter CRConvNet for Character Recognizer ConvNet, takes as input a color image of a character mapped into 3 $T \times T$ maps, one map for each color channel, and containing values normalized between -1 and 1, and returns a vector of N values (with N the number of classes of characters) where each value (between -1 and 1) encodes a score of belonging to a given class of characters. The ConvNet architecture contains four convolutional layers and two neural layers (*cf.* Fig. 5). The first two layers (a convolution followed by a sub-sampling layer) can be interpreted as a feature extractor where the sub-sampling layer permits to reduce sensitivity to affine transformations and to reduce computational complexity. The two next layers (a convolution followed by a sub-sampling layer) combine extracted

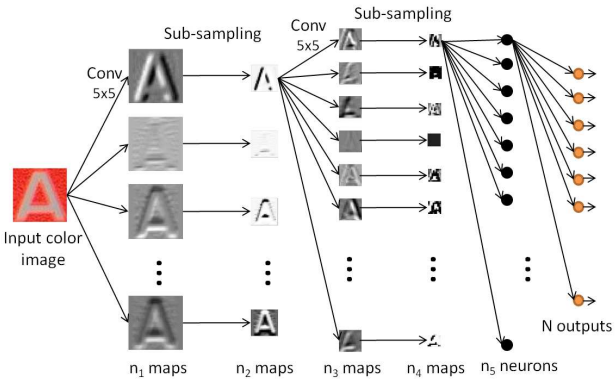

Fig. 5 CRConvNet: the Character Recognition Convolutional Neural Network architecture.

features considering their spatial relationships. The two last layers correspond to a classical multilayer perceptron and provide output scores that can be interpreted as the probabilities of the input image to belong to each class of character. CRConvNet is trained with the classical back-propagation algorithm with momentum.

3.3 Text recognition

Individual character recognition results can now be combined in order to determine the whole text present in images or in videos. Since text images are segmented into separated characters, we intuitively recognize texts as the sequences of recognized characters. In subsection 3.1, two categories of segmentation borders were distinguished: "accurate" and "risky" ones. Characters located between two "accurate" segmentations (green separations in Fig. 6) are recognized and directly considered as letters of the text. However when, between two successive "accurate" segmentations, "risky" ones are observed (red separations in Fig. 6), the CRConvNet is applied on each possible segmentation ("accurate" and "risky") and the configuration that obtains the highest score is selected. Figure 6 illustrates all the candidate characters on which CRConvNet is applied. At this stage of the processing, for each possible segmentation, only the best response (*i.e.*, the class obtaining the highest probability) of the CRConvNet is considered while the rest of the responses is ignored.

As shown in Fig. 6, even though errors related to "risky segmentations" are reduced, confusions between similar characters are still present (such as the "v" recognized as a "y"). In section 5, we show how to introduce linguistic knowledge able to drive the recognition scheme and to tackle these character confusions.

Fig. 6 An example of recognized texts: green arcs illustrate characters located between successive "accurate" segmentations and red arcs represent different possible configurations related to "risky segmentations".

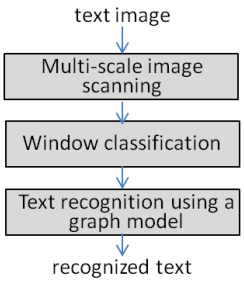

Fig. 7 Outline of the segmentation-free OCR.

4 The segmentation-free OCR

Our first OCR system relies on the segmentation of text images into individual characters. However in the case of text images with strong distortions, this step can lead to potential under- or over-segmentation. Inspired by works dedicated to handwritten recognition [9,14], which use a sliding window technique, we propose a second OCR approach able to avoid this segmentation step by incorporating a multi-scale scanning process [10]. The different steps of this OCR are presented in Fig. 7 and described in this section.

4.1 Multi-scale image scanning

The first step of our segmentation-free OCR consists in scanning the text image. This is done by moving a sliding window, from the left to the right, centered at regular and close positions. In our experiments, best results were obtained with a moving step of one eighth of the image height h. We also consider windows at various scales (*i.e.*, windows with different widths) in order to cover different character sizes. Four scales (namely S_1, S_2, S_3 and S_4) are used in our experiments, corresponding to window widths equal to $\frac{h}{4}$, $\frac{h}{2}$, $\frac{3h}{4}$, and h. Figure 8 illustrates an example of a text image scanned at different scales and shows characters framed at their corresponding scales (*e.g.*, "P" and "e" are framed with windows equal to h and $\frac{h}{2}$ respectively) and an example of a misaligned window. A classification is applied to every window to identify non valid characters and recognize valid ones.

Fig. 8 Multi-scale image scanning process: examples of characters well framed at scales S_4 (red), S_2 (blue), S_3 (orange), and S_1 (green) and a misaligned window at scale S_2 (blue).

Fig. 9 Non-linear borders of multi-scale sliding windows.

In order to improve recognition performance of characters framed with sliding windows, we propose to adapt the window borders to the local morphology of the images. The purpose is to remove parts of other characters that can be extracted with the central character when using vertical borders of windows (*e.g.*, the character "o" extracted with the character "P" in Fig. 8). At each window position and scale, non-linear borders are defined as shortest vertical paths within the text image as described in subsection 3.1. In case of important image distortions, non separated characters or misaligned windows, the shortest path algorithm induces straight vertical borders since pixels in the local area have the same probability. Figure 9 shows some examples of the obtained windows with non-linear borders and gives an example of straight vertical borders due to the non separation between two characters ("S" and "T" in the word "STAR" in Fig. 8).

4.2 Window classification

Before recognizing characters, a step of pre-sorting is required to identify windows containing "valid" characters and "non valid" ones. Hence, we propose to use a Convolutional Neural Network whose task is to classify windows as "valid character" or "garbage" (*i.e.*, window misaligned with a character, part of a character or interstice between characters).

In our application, several network architectures were tested. The best configuration, hereafter WCConvNet for Window Classifier ConvNet, takes as input a color window image mapped into three $T \times T$ input maps, containing values normalized between -1 and 1. The architecture of WCConvNet is similar to that of CR-ConvNet presented in subsection 3.2 except that it has a single output neuron trained to respond -1 for "garbage" windows, and $+1$ for "valid" characters. After training, windows obtaining a negative output are labeled as "garbage", while the others are presented to the CR-ConvNet (see subsection 3.2). Figure 10 illustrates the

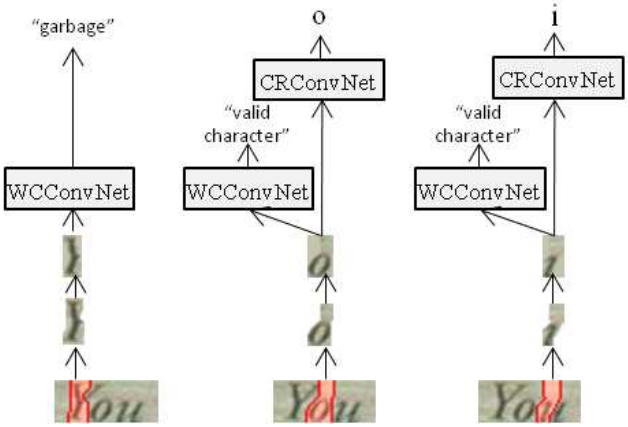

Fig. 10 Window classification process.

whole window classification scheme: interstices between characters are labeled as "garbage" and well framed characters (such as "o") are recognized. Nevertheless, some parts of characters produce recognition confusions (*e.g.*, the part of "u" on the right in Fig. 10 is recognized as an "i"). In the next subsection, we show how we deal with window classification results and handle recognition errors.

4.3 Text recognition using a graph model

Multi-scale window classification results can now be combined to recognize texts extracted from images or videos. To that end, we choose to use a directed acyclic graph model able to represent the spatial constraints between different overlapping windows (*cf.* Fig. 11). The borders of the windows are represented by vertices. The first (resp. last) vertex in the graph corresponds to the left (resp. right) border of the first (resp. last) window position in the text image. Vertices are connected by directed edges, called arcs, each representing one extracted window. Since the multi-scale scanning scheme includes four different window sizes, each vertex v is thus connected to four successor vertices (*i.e.*, the right borders of the four different windows starting from v).

In subsection 4.1, we have explained how non-linear borders of windows are computed and characterized by scores encoding their probabilities to correspond to borders between characters. These segmentation scores are assigned to their corresponding vertices in the graph. In the same way, the classification results of each window (*i.e.*, the output of WCConvNet for non valid characters and the best output of CRConvNet for the valid ones) are assigned to each arc. Figure 11 shows one part of the graph built on a sample image representing each possible window. A best path search algorithm, namely

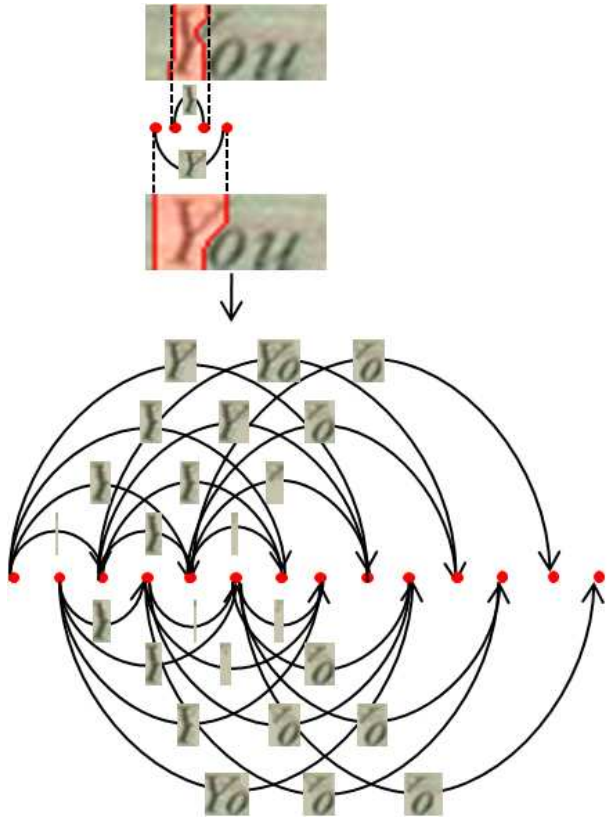

Fig. 11 Graph model recognition.

the classical Viterbi algorithm, is then applied to determine the best sequence of characters corresponding to the text image. All possible paths within the built graph can then be tested and evaluated taking into account the spatial constraints between sliding windows, the scores of windows borders and the classification results. The recognized text is thus obtained as the sequence of characters corresponding to the path having the best score and avoiding arcs which represent windows that do not contain valid characters.

5 Improving recognition results by integrating linguistic information

Both proposed OCR systems still generate some errors related to character confusions and incorrect segmentations or misaligned windows. To tackle these ambiguities and improve the recognition rate, we propose to incorporate into both OCRs some information provided by the lexical context. The idea is to take advantage of the language properties and to introduce some linguistic knowledge in order to supervise the recognition process.

In this context, n-gram models (widely used in speech recognition) have shown to be well adapted to our recog-

nition problem [1]. By learning the joint probabilities of sequences of items (which can be phonemes, words or characters), these models allow to predict the next item to be recognized given items that have just been recognized. For our recognition problem, since single words or short sentences are considered, a character n-gram model is trained over a corpus of words to estimate the sequences of letters probabilities in a given language. These probabilities are then integrated into both recognition frameworks to adjust transitions between characters. Hence, when evaluating word scores in the segmentation-based OCR or path scores within the graph of the segmentation-free OCR, these joint probabilities of character sequences are introduced to weight the different word propositions and paths. Furthermore for each recognition result, not only the best class of character—namely the class obtaining the best output value—is considered but also the next four best classes. Thus new word propositions can be tested and evaluated aiming to reduce character confusion (*e.g.*, the "v" recognized as a "y" in Fig. 6).

Typically, for a given text image, if we assume that $\hat{C}$ is the sequence of characters present in the image, $\hat{C}$ can be characterized as the sequence which maximizes the probability $p(C|sig)$, where C is a sequence of characters and sig is the given text image. A Maximum A Posteriori (MAP) approach is applied as follows:

$$\hat{C} = \underset{C}{\operatorname{argmax}}\,(p(C|sig)) = \underset{C}{\operatorname{argmax}}\,(p(sig|C)p(C)) \quad (3)$$

where $p(sig|C)$ is the a posteriori probability of sig given the character sequence C, and $p(C)$ is the a priori probability of C. $p(sig|C)$ is computed from the character (or window) recognition outputs as follows:

$$p(sig|C) = \prod_i p(s_i|c_i) \quad (4)$$

where c_i and s_i are respectively the ith character of C and its image (or window). $p(C)$ is obtained from the language model. Using the n-gram model, we assume that a character only depends on its $n-1$ predecessors; thus:

$$p(C) = \prod_i p(c_i|\phi(h_i)) = \prod_i p(c_i|c_{i-n+1}...c_{i-1}) \quad (5)$$

where $\phi(h_i)$ is the context of a character c_i and corresponds to the sequence $c_{i-n+1}...c_{i-1}$. In our previous work [11], the best recognition results were obtained using a character tri-gram model (n=3). Hence we chose to use this model in our experiments.

Because probabilities are low (between 0 and 1), their decimal logarithms (between $-\infty$ and 0) are preferred, and two coefficients γ and δ are introduced as follows:

$$\hat{C} = \arg max \sum_i (log(p(s_i|c_i)) + \gamma log(p(c_i|\phi(h_i))) + \delta) \quad (6)$$

γ, called the Grammar Scale Factor, encodes the weight of the language model and serves to balance the influence of the linguistic knowledge in the recognition process, while δ is incorporated to compensate the over- and under-segmentations by controlling lengths of word candidates.

In our experiments, using the *SRILM* toolkit [40], two language models, one for the English language and another for the French language, were trained on two corpora of about respectively $11,000$ English words and $10,000$ French words.

6 Experimental setup and results

This section reports two main experiments, compares the proposed OCRs and discusses their results. After a presentation of the datasets used in the experiments, the proposed OCR systems are evaluated and compared to other state-of-the-art approaches. Results are also analyzed highlighting the benefits and the limits of the character segmentation step. The contribution of the language model incorporated in both recognition schemes is also evaluated.

6.1 Text image datasets

Our experiments have been carried out on two types of multimedia documents: "caption texts" in videos, and "scene text" images.

6.1.1 The "caption text" video dataset (Dataset I)

This dataset consists of 12 videos of French news broadcast programs. Each video, encoded in MPEG-4 (H. 264) format at 720×576 resolution, is about 30 minutes long. In this paper, we focus on text recognition; however since the first task for video text recognition consists in detecting and extracting texts from videos, this step is performed using the text detector proposed by Delakis *et al.* [6] as described in [11]. Each video contains about 400 words, roughly corresponding to 2,200 characters (*i.e.*, small and capital letters, numbers and punctuation marks). As shown in Fig. 12, embedded texts can vary a lot in terms of size (a height of 8 to 24 pixels), color, style and background (uniform and complex backgrounds).

Four videos of this dataset are used to generate a database of $15,168$ images of single character perfectly segmented and $1,001$ images of non valid characters (*i.e.*, "garbage"). The obtained database, called CharDataset I, is used to train WCConvNet and CRConvNet. The other eight videos, called TextDataset I, are

Fig. 12 Examples of caption texts extracted from videos.

Fig. 13 Examples of scene text images from the ICDAR 2003 database.

annotated to evaluate the OCRs' recognition performance. In this dataset, 41 character classes are considered: 26 Latin letters, 10 Arabic numbers, 4 special characters ('.', '-', '(', and ')'), and a class for spaces between words.

6.1.2 The "scene text" dataset (Dataset II)

This dataset is the public database ICDAR 2003[1] created for a competition on scene word recognition [25]. It contains English scene text images of different sizes (a height of 12 to 504 pixels), presents several kinds of distortions (non uniform illumination, occlusions, blur, etc.) and contains characters printed, written and painted in various fonts and colors (*cf.* Fig. 13).

The ICDAR 2003 database consists of two distinct databases: an isolated character database of 5,689 images of characters and a scene text database of 2,266 images of text in which 1,156 images are provided for training and 1,110 images for test. The training set of text images is used to generate 4,056 images of non valid characters that we add to the single character database to obtain a set of 9,745 images, called CharDataset II, used to train WCConvNet and CRConvNet. The test set, called TextDataset II, is used to evaluate the two proposed OCRs. For the "scene text" dataset, 36 classes of characters are considered: 26 Latin letters and 10 Arabic numbers.

[1] The database ICDAR 2003 is available for download at http://algoval.essex.ac.uk/icdar/Datasets.html#Robust.

Table 1 Classification performance of WCConvNet and CRConvNet.

	CharDataset I	CharDataset II
WCConvNet	87.99%	79.23%
CRConvNet	98.04%	85.13%

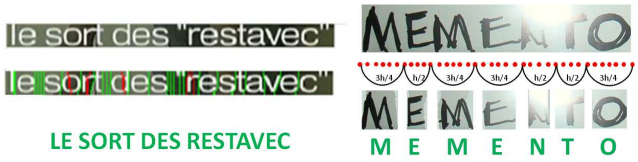

Fig. 14 Examples of recognized texts: the text on the left is recognized using the segmentation-based OCR and the text on the right is recognized using the segmentation-free OCR.

6.2 ConvNets trainings and results

In both experiments, called "caption text" and "scene text" recognition, CharDataset I and CharDataset II are divided into two subsets: one containing 90% of the images and used to train WCConvNet and CRConvNet, and another set containing the remaining 10% used to evaluate classification performance and generalization. Table 1 shows the classification results on both subsets. A 10% difference for both classifiers (WRConvNet and CRConvNet) between the results obtained on CharDataset I and CharDataset II can be noticed. This difference can be explained by the high variability of "scene text" characters compared to "caption text" characters. CRConvNet usually obtains better performance than WCConvNet for which the classification task seems to be harder because of important confusions between some mis-segmented characters (considered as non valid ones) and other valid characters, such as a part of a "u" that can be recognized as an "i".

6.3 Performance of the proposed OCRs

Using the trained ConvNets, the segmentation-based and segmentation-free OCRs are tested and evaluated both on TextDataset I and TextDataset II. Figure 14 presents examples of texts recognized using these OCRs and illustrates the character segmentation step of the first OCR and the resulting best path within the graph model of the second OCR. Recognition results are reported in Table 2.

Experiments carried out on TextDataset I show that both OCRs perform well on embedded texts with more than 93% of good character recognition rate and that they obtain similar results (less than 2% of difference in term of character recognition rate). However, the segmentation-based OCR is slightly better than the segmentation-free OCR on this dataset. This proves

that when the character segmentation step works well (on text with small distortion like "caption text"), it enhances the following steps and leads to better recognition performance. Particularly, the segmentation-based OCR system obtains higher recognition rate on spaces between words than the second OCR. On the contrary, results obtained on natural scene texts (*i.e.*, TextDataset II) show a larger difference between OCRs' performance. While the segmentation-free OCR achieves a character recognition rate above 70%, corresponding to a word recognition rate of about 47%, the segmentation-based OCR achieves 65% of character recognition rate corresponding to a word recognition rate of 41%. These results demonstrate the drawbacks of the segmentation step where any error directly induces a drop in the recognition accuracy. Particularly, in the case of natural scene text, images are usually affected by various distortions which make the segmentation very hard and thus lead to errors such as false segmentations considered as "accurate" ones that over-segment characters, or to confusing "risky" segmentations.

The difference between the performances achieved on TextDataset II and those obtained on TextDataset I can be explained by the fact that: (i) the character recognizer performs better for CharDataset I, and (ii) TextDataset I is less complex than TextDataset II. In addition, the few remaining errors on TextDataset I can be justified by some character confusions between visually similar characters and some character segmentation errors in the case of the segmentation-based OCR. Regarding the errors produced on TextDataset II, the strong distortions of an important number of images and the small sizes of some of them are the major causes of the remaining errors (30% of characters in the case of the segmentation-free OCR and 35% in the case of the segmentation-based OCR).

Concerning the computational time, the segmentation-based OCR is in average 7 times faster than the second OCR. For instance, for a text image with a size of 418 pixels, while the segmentation-based OCR takes 700 ms the segmentation-free OCR takes 5000 ms.

Table 3 presents a comparison of the two proposed OCRs with state-of-the-art methods [34, 42] and commercial OCR engines (ABBYY FineReader OCR and Tesseract OCR). Notice that ABBYY FineReader OCR and Tesseract OCR were not trained on the same datasets as our OCRs and other state-of-the-art methods. This fact is due to practical issues: actually the training of ABBYY OCR requires to purchase the SDK that we do not own and the last version of Tesseract OCR (namely Tesseract 3.0x) is not adapted to deal with real text images.

Since Saïdane *et al.* [34] and Wang *et al.* [42] have designed their methods to recognize single words in natural scene images, comparisons with previously published state-of-the-art methods is done only on the public database ICDAR 2003 (the "caption text" video dataset contains mainly images with sentences). In these comparisons, three experiments were performed on TextDataset II, evaluating the word recognition rate as in [34] and [42]. Besides the experimentation on the full TextDataset II (Exp1), the OCRs are evaluated on the 901 images selected in [34] (Exp2) and on the $1,065$ images selected in [42] using the same lexicon, created from all the words that appear in the test set, as the one used in [42] (Exp3). These different tests show that our segmentation-free OCR yields the best word accuracy. It also outperforms Wang *et al.*'s system [42] by about +7% even though their method uses hand-designed features to recognize characters.

Concerning commercial OCR systems, namely ABBYY FineReader and Tesseract, since they were not trained on the same datasets as those used for our systems, results reported in Table 3 are basically provided for illustrative purposes. Experiments carried out on "caption text" in TextDataset I show that Tesseract OCR obtains a word recognition rate of 70% while ABBYY FineReader obtains a word recognition rate of 87%. Even though ABBYY FineReader is not trained, it still outperforms our system. In our opinion, this is due to the use of a dictionary, absent in our OCRs (experiments with a small dictionary enabled us to obtain equivalent performances); we also notice some extra errors in our systems due to a confusion between quotes (which are not considered in our CRConvNet) and the letter "l", inducing word errors. In the case of "scene text" in TextDataset II, ABBYY FineReader OCR and Tesseract OCR evaluated by Wang *et al.* [42] obtain poor results with less than 45% of word recognition rate while our segmentation-free OCR achieves 66%. Hence, our OCR systems prove their great ability to handle both "caption" and "scene texts".

6.4 Contribution of the language model

This subsection focuses on the contribution of the language model incorporated in our OCR systems. Recognition performance of both OCRs integrating language models (presented in subsection 6.3) is compared to their performance when no linguistic knowledge is provided.

Table 4 highlights the contribution of the language model (LM) integrated into the designed OCRs and evaluated on TextDataset I and TextDataset II. All

Table 2 Recognition performance of the proposed OCR systems; RR means recognition rate.

OCR system	TextDataset I		TextDataset II	
	Character RR	Word RR	Character RR	Word RR
Segmentation-based OCR	95.56%	85.80%	65.33%	41.19%
Segmentation-free OCR	93.55%	81.32%	70.33%	46.72%

Table 3 Comparison of the proposed OCR systems to state-of-the-art methods and commercial OCR engines (Note that ABBYY FineReader OCR and Tesseract OCR were not trained on the same datasets as the other systems).

OCR system	TextDataset I		TextDataset II (Word RR)		
	Character RR	Word RR	Exp1	Exp2	Exp3
Segmentation-based OCR	95.56%	85.80%	41.19%	-	-
Segmentation-free OCR	93.55%	81.32%	46.72%	57.04%	66.19%
Saïdane *et al.* [34]	-	-	-	54.13%	-
Wang *et al.* [42]	-	-	-	-	59.20%
ABBYY FineReader OCR	95.03%	87.70%	-	-	42.80%
Tesseract OCR	88.57%	70.01%	-	-	35.00%

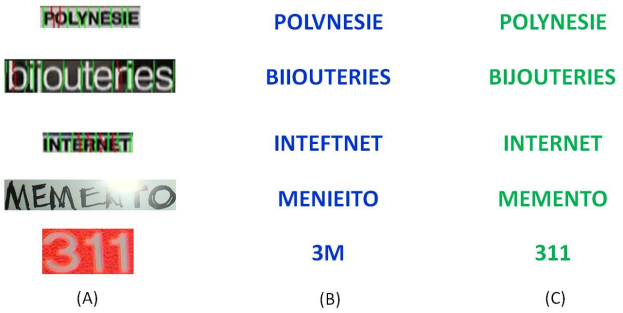

Fig. 15 Examples of text images recognized by the proposed OCRs (the three first results are obtained with the segmentation-based OCR and the two others with the segmentation-free OCR): (A) text images, (B) results before integrating the language model and (C) results after the integration of the tri-gram language model.

performed experiments demonstrate that the integration of the character tri-gram language model results in an important improvement on the character recognition rate, which reaches 16% in the case of "scene texts" recognized by the segmentation-free OCR. The language model also increases the word recognition rate considerably, by +13% in average.

Figure 15 illustrates some confusions corrected using the language model, especially for similar characters such as "i" and "j", and shows an example of discarded over-segmentation, namely "r" previously over-segmented into "f" and "t".

7 Conclusion and future work

In this paper, we have presented two different approaches for text recognition in multimedia documents (images and videos). The first OCR relies on a step of character segmentation that aims at separating characters before recognizing them. One of the contributions of this system is the non-linear segmentation performed in order to obtain borders well-adapted to the local morphology of text images and thus improve recognition rates. In contrast, the second OCR avoids character segmentation by using a multi-scale scanning scheme and a graph model. Unlike other state-of-the-art methods, this system allows to recognize characters at their appropriate positions and scales directly from the whole text image without any pre-processing. A robust neural-based classification method is designed to recognize characters and is used in both OCR systems. Linguistic knowledge is also integrated in both systems to remove errors.

Both systems were tested and evaluated on texts embedded in videos and on natural scene text images. Our experiments showed that both OCRs perform very well (over 93% of correctly recognized characters) in the case of "caption text" images. In the case of images with strong distortions, like natural scene texts, the segmentation-free OCR performs well, achieving good results (of about 70% of character recognition rate) better than those provided with the segmentation-based OCR. The proposed OCRs were also compared to other state-of-the-art methods and obtained the best results. In this paper, we have also confirmed that the incorporation of linguistic knowledge, namely a character n-gram model, improves performance of both OCRs.

Their high efficiency allows to use our OCRs in automatic indexing and retrieval systems, like TV news broadcast content analysis. Moreover, the genericity of our systems permits to use them in many applications. For instance, they can serve to enhance a video teaching service by recognizing texts embedded in filmed slides, or help visually impaired people by reading using audio devices.

As a future extension of this work, we plan to use an unsupervised learning technique (namely autoencoders) to produce relevant representations of text images. A

Table 4 Contribution of the language model incorporated in the proposed OCR systems; LM means language model.

OCR system	TextDataset I		TextDataset II	
	Character RR	**Word RR**	**Character RR**	**Word RR**
Segmentation-based OCR without LM	88.14%	63.04%	61.12%	34.75%
Segmentation-based OCR	95.56%	85.80%	65.33%	41.19%
Segmentation-free OCR without LM	72.30%	24.00%	54.32%	25.54%
Segmentation-free OCR	93.55%	81.32%	70.33%	46.72%

connectionist neural model can then be trained to recognize the encoded sequences of characters.

References

1. Bahl, L., Brown, P., de Souza, P., Mercer, R.: A tree-based statistical language model for natural language speech recognition. IEEE Transactions on Acoustics, Speech and Signal Processing **37**(7), 1001–1008 (2002)
2. Casey, R., Lecolinet, E.: A survey of methods and strategies in character segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(7), 690–706 (2002)
3. Chen, D., Odobez, J., Bourlard, H.: Text detection and recognition in images and video frames. Pattern Recognition **37**(3), 595–608 (2004)
4. Chen, T., Ghosh, D., Ranganath, S.: Video-text extraction and recognition. In: IEEE Region 10 Conference, TENCON'04, vol. 1, pp. 319–322 (2005)
5. Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D., Ng, A.: Text detection and character recognition in scene images with unsupervised feature learning. In: International Conference on Document Analysis and Recognition, pp. 440–445 (2011)
6. Delakis, M., Garcia, C.: Text detection with convolutional neural networks. In: International Conference on Computer Vision Theory and Applications, vol. 2, pp. 290–294 (2008)
7. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) **39**(1), 1–38 (1977)
8. Dorai, C., Aradhye, H., Shim, J.C.: End-to-end video text recognition for multimedia content analysis. In: International Conference on Multimedia and Expo, pp. 601–604 (2001)
9. El Abed, H., Margner, V.: Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten Arabic words. In: International Conference on Document Analysis and Recognition, vol. 2, pp. 974–978 (2007)
10. Elagouni, K., Garcia, C., Mamalet, F., Sébillot, P.: Combining multi-scale character recognition and linguistic knowledge for natural scene text OCR. In: International Workshop on Document Analysis Systems, pp. 120–124 (2012)
11. Elagouni, K., Garcia, C., Sébillot, P.: A comprehensive neural-based approach for text recognition in videos using natural language processing. In: International Conference on Multimedia Retrieval (2011)
12. 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), 1408–1423 (2004)
13. Halima, M., Karray, H., Alimi, A.: A comprehensive method for arabic video text detection, localization, extraction and recognition. Advances in Multimedia Information Processing, pp. 648–659 (2011)
14. Hamdani, M., El Abed, H., Kherallah, M., Alimi, A.: Combining multiple HMMs using on-line and off-line features for off-line Arabic handwriting recognition. In: International Conference on Document Analysis and Recognition, pp. 201–205 (2009)
15. Hua, X., Yin, P., Zhang, H.: Efficient video text recognition using multiple frame integration. In: International Conference on Image Processing, vol. 2, pp. 397–400 (2002)
16. Jung, K., In Kim, K., K Jain, A.: Text information extraction in images and video: a survey. Pattern Recognition **37**(5), 977–997 (2004)
17. Kopf, S., Haenselmann, T., Effelsberg, W.: Robust character recognition in low-resolution images and videos. Universität Mannheim/Institut für Informatik (2005)
18. Kusachi, Y., Suzuki, A., Ito, N., Arakawa, K.: Kanji recognition in scene images without detection of text fields-robust against variation of viewpoint, contrast, and background texture. In: International Conference on Pattern Recognition, vol. 1, pp. 457–460 (2004)
19. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks pp. 255–258 (1995)
20. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. of the IEEE **86**(11), 2278–2324 (1998)
21. Li, H., Doermann, D., Kia, O.: Automatic text detection and tracking in digital video. IEEE Transactions on Image Processing **9**(1), 147–156 (2000)
22. Li, M., Bai, M., Wang, C., Xiao, B.: Conditional random field for text segmentation from images with complex background. Pattern Recognition Letters **31**(14), 2295–2308 (2010)
23. Lienhart, R., Stuber, F.: Automatic text recognition in digital videos. In: Proc of SPIE Image and Video Processing IV, vol. 2666, pp. 180–188 (1996)
24. Lim, J., Park, J., Medioni, G.: Text segmentation in color images using tensor voting. Image and Vision Computing **25**(5), 671–685 (2007)
25. Lucas, S., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: ICDAR 2003 robust reading competitions. International Journal on Document Analysis and Recognition **2**, 682–687 (2003)
26. Mancas-Thillou, C., Gosselin, B.: Character segmentation-by-recognition using log-Gabor filters. In: International Conference on Pattern Recognition, vol. 2, pp. 901–904 (2006)
27. Miao, G., Zhu, G., Jiang, S., Huang, Q., Changsheng, X., Gao, W.: A real-time score detection and recognition approach for broadcast basketball video. In: International Conference on Multimedia and Expo, pp. 1691–1694 (2007)

28. Mishra, A., Alahari, K., Jawahar, C.: An MRF model for binarization of natural scene text. In: International Conference on Document Analysis and Recognition, pp. 11–16 (2011)
29. Negishi, K., Iwamura, M., Omachi, S., Aso, H.: Isolated character recognition by searching features in scene images. In: International Workshop on Camera-Based Document Analysis and Recognition, pp. 140–147 (2005)
30. Ntirogiannis, K., Gatos, B., Pratikakis, I.: Binarization of textual content in video frames. In: International Conference on Document Analysis and Recognition, pp. 673–677 (2011)
31. Phan, T., Shivakumara, P., Su, B., Tan, C.: A gradient vector flow-based method for video character segmentation. In: International Conference on Document Analysis and Recognition, pp. 1024–1028 (2011)
32. Saïdane, Z., Garcia, C.: Automatic scene text recognition using a convolutional neural network. In: Conference on Computer Vision and Pattern Recognition, pp. 100–106 (2007)
33. Saïdane, Z., Garcia, C.: Robust binarization for video text recognition. In: International Conference on Document Analysis and Recognition, vol. 2, pp. 874–879 (2007)
34. Saïdane, Z., Garcia, C., Dugelay, J.: The image text recognition graph (iTRG). In: International Conference on Multimedia and Expo, pp. 266–269 (2009)
35. Sato, T., Kanade, T., Hughes, E., Smith, M., Satoh, S.: Video OCR: indexing digital news libraries by recognition of superimposed captions. Multimedia Systems **7**(5), 385–395 (1999)
36. Sharma, N., Pal, U., Blumenstein, M.: Recent advances in video based document processing: A review. In: International Workshop on Document Analysis Systems, pp. 63–68 (2012)
37. Shivakumara, P., Bhowmick, S., Su, B., Tan, C., Pal, U.: A new gradient based character segmentation method for video text recognition. In: International Conference on Document Analysis and Recognition, pp. 126–130 (2011)
38. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for Convolutional Neural Networks applied to visual document analysis. In: International Conference on Document Analysis and Recognition, vol. 2, pp. 958–963 (2003)
39. Som, T., Can, D., Saraclar, M.: HMM-based sliding video text recognition for Turkish broadcast news. In: International Symposium on Computer and Information Sciences, pp. 475–479 (2009)
40. Stolcke, A.: SRILM-An extensible language modeling toolkit. In: International Conference on Spoken Language Processing, vol. 3, pp. 901–904 (2002)
41. Wakahara, T., Kita, K.: Binarization of color character strings in scene images using K-means clustering and support vector machines. In: International Conference on Document Analysis and Recognition, pp. 274–278 (2011)
42. Wang, K., Belongie, S.: Word spotting in the wild. In: European Conference on Computer Vision, pp. 591–604 (2010)
43. Weinman, J., Learned-Miller, E., Hanson, A.: Scene text recognition using similarity and a lexicon with sparse belief propagation. Pattern Analysis and Machine Intelligence **31**(10), 1733–1746 (2009)
44. Yager, R.: Connectives and quantifiers in fuzzy sets. Fuzzy Sets and Systems **40**(1), 39–75 (1991)
45. Yamazoe, T., Etoh, M., Yoshimura, T., Tsujino, K.: Hypothesis preservation approach to scene text recognition with weighted finite-state transducer. In: International Conference on Document Analysis and Recognition, pp. 359–363 (2011)
46. Ye, Q., Huang, Q., Gao, W., Zhao, D.: Fast and robust text detection in images and video frames. Image and Vision Computing **23**(6), 565–576 (2005)
47. Yi, J., Peng, Y., Xiao, J.: Using multiple frame integration for the text recognition of video. In: International Conference on Document Analysis and Recognition, pp. 71–75 (2009)
48. Yokobayashi, M., Wakahara, T.: Segmentation and recognition of characters in scene images using selective binarization in color space and GAT correlation. In: International Conference on Document Analysis and Recognition, pp. 167–171 (2005)
49. Zhang, D., Chang, S.: A Bayesian framework for fusing multiple word knowledge models in videotext recognition. In: Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 528–533 (2003)
50. Zhou, Z., Li, L., Tan, C.: Edge based binarization for video text images. In: International Conference on Pattern Recognition, pp. 133–136 (2010)