Pattern Anal Applic (2013) 16:431–446
DOI 10.1007/s10044-011-0237-7

INDUSTRIAL AND COMMERCIAL APPLICATION

Detection of artificial and scene text in images and video frames

Marios Anthimopoulos · Basilis Gatos · Ioannis Pratikakis

Received: 6 August 2010 / Accepted: 2 September 2011 / Published online: 23 September 2011

Abstract Textual information in images and video frames constitutes a valuable source of high-level semantics for multimedia indexing and retrieval systems. Text detection is the most crucial step in a multimedia text extraction system and although it has been extensively studied the past decade still, it does not exist a generic architecture that would work for artificial and scene text in multimedia content. In this paper we propose a system for text detection of both artificial and scene text in images and video frames. The system is based on a machine learning stage which uses an Random Forest classifier and a highly discriminative feature set produced by using a new texture operator called Multilevel Adaptive Color edge Local Binary Pattern (MACeLBP). MACeLBP describes the spatial distribution of color edges in multiple adaptive levels of contrast. Then, a gradient-based algorithm is applied to achieve distinction among text lines as well as refinement in the localization of the text lines. The whole algorithm is situated in a multiresolution framework to achieve invariance to scale for the detection of text lines. Finally, an optional connected-component step segments text lines into words based on the distances between the resulting components. The experimental results are produced by applying a concise evaluation methodology and prove the superior performance achieved by the proposed text detection system for artificial and scene text in images and video frames.

Keywords Text detection · Artificial text · Scene text · Natural scene images · Video OCR · Multimedia information retrieval

1 Introduction

The proliferation of high-performance low-priced digital cameras embedded in mobile devices, combined with today's internet sharing capabilities has caused an outbreak of available multimedia content. Television broadcasting through internet has also contributed towards this direction. Huge digital libraries have been created raising the need for information extraction and indexing. This trend requires from the information retrieval systems to overcome the challenges and adapt to the new kind of data. Textual information in multimedia constitutes a very rich source of high-level semantics for retrieval and indexing. Document image processing, after many decades of research, has reached a high level of text recognition accuracy, for traditional scanner-based images. However, these techniques fail to deal with text appearing in videos or camera-based images.

Mainly, there exist two kinds of text occurrences in videos and images, namely artificial and scene text. Artificial text, as the name implies, is artificially added to describe the multimedia content or give additional information related to it. Scene text is textual content that was captured by a camera as part of a scene, e.g. text on T-shirts or road signs. Moreover, artificial text usually refers to text

M. Anthimopoulos (✉) · B. Gatos
Computational Intelligence Laboratory,
Institute of Informatics and Telecommunications,
National Center for Scientific Research "Demokritos",
153 10 Athens, Greece
e-mail: anthimop@iit.demokritos.gr

B. Gatos
e-mail: bgat@iit.demokritos.gr

I. Pratikakis
Department of Electrical and Computer Engineering,
Democritus University of Thrace, 671 00 Xanthi, Greece
e-mail: ipratika@ee.duth.gr

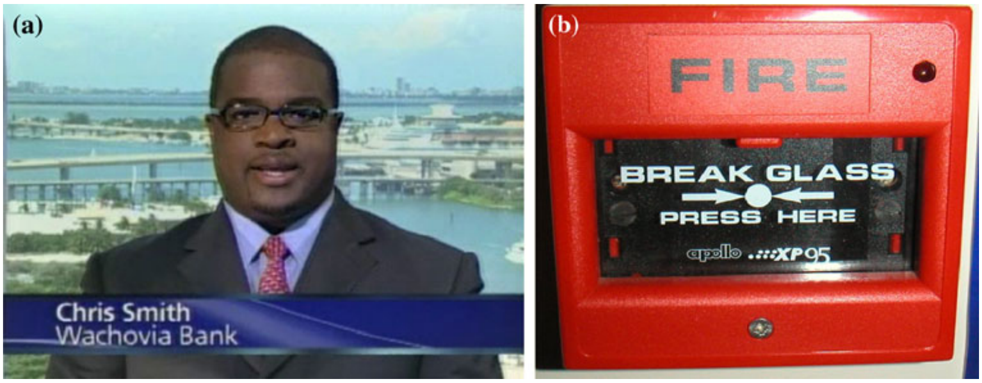


Fig. 1 **a** Video frame with artificial text. **b** Image with scene text

embedded in videos while scene text refers mainly to natural scene images. Figure 1a presents a video frame with artificial text and Fig. 1b a natural scene image containing scene text. Text can also be classified into normal or inverse. Normal is denoted any text whose characters have lower intensity values than the background while inverse text is the opposite. In Fig. 1b, "FIRE" is considered as normal while the rest of the text is considered inverse.

Several multimedia text extraction methods have been proposed the last decade. Some of them focus on artificial text in videos while others deal with natural scene text in camera-based images. Each of these kinds of text has distinct characteristics and brings new challenges to the research area compared to classic flatbed scanned documents. Methods designed for artificial video text assume horizontal text lines with multi-frame occurrence. However, this kind of text often suffers from low resolution and compression noise. Video frames may have even lower resolution than VGA (640 × 480) when scanners often produce images larger than 2480 × 3508. Moreover, video compression affects high frequencies, inserting noise and creating colour leaking phenomena. This fact may smooth character edges or even distort text colour homogeneity which constitutes fundamental features of text. On the other hand, scene text of natural images usually suffers from uneven lighting, perspective distortion and image blurring. Arbitrary natural lighting combined with artificial light from camera's flash can produce reflections or shadows while 3D distortion of text can cause severe problems in both text detection and recognition. Image blurring caused by abrupt motion or wrong camera focus may have effects similar to video compression eliminating typical text features like edges. One common problem of both artificial and scene text in images and videos is the complex background. Contrary to traditional scanned documents, the background of colour images and videos is not clean but may contain objects like building windows or tree leaves that constitute a text-like texture.

The remainder of this paper is organized as follows: Sect. 2 outlines the related work; Sect. 3 presents the proposed text detection system; Sect. 4 provides the related experimental results and finally Sect. 5 concludes the paper.

2 Related work

Although artificial text and scene text extraction are often considered as different topics because of their distinct peculiarities, they actually refer to the same research area since they meet the same main challenge of discriminating text areas from non-text complex backgrounds. In this section we will outline the techniques found in literature for spatial text detection of both artificial and scene text in images and video frames.

In general, the existing text detection methods can be roughly divided in two categories: region-based and texture-based. Region-based methods group pixels that belong to the same character based on the colour homogeneity, the strong edges between character and background or by using a stroke filter. Then, the detected characters are grouped to form text lines according to colour, size and geometrical rules. Texture-based algorithms scan the image at different scales using a sliding window and classify image areas as text or non-text based on texture-like features. Another possible categorization of text detection methods could be dividing them into heuristic and machine learning techniques. Usually region-based algorithms apply heuristic rules to group pixels into characters and then to text lines, while texture-based approaches rely on machine learning classifiers trained on real data for the discrimination between text and non-text areas. However, despite all possible categorization of reported methods, there are many works in literature that use both region-based and texture-based approaches to exploit all existing information or even combine machine learning techniques with heuristic rules trading off between accuracy and efficiency.

Many heuristic, region-based methods, derived from document analysis research area, are based on color or intensity homogeneity of characters. They detect character regions in the image and then group them into words and text lines based on geometrical constraints. These methods, also known as connected component (CC) methods, can perform satisfactorily only on high quality images with simple background and known text color. However, these assumptions rarely apply in real cases since text in video images often suffers from color bleeding due to video compression while in natural images uneven lighting conditions may spoil the constant text color. Moreover, when background is really complex, connected component analysis becomes computationally expensive. Typical CC approaches for text in videos and book covers can be found, respectively, in [1] and [2]. Wang et al. [3] propose a CC method with recognition feedback for the detection of Asian characters in scene images.

Some other heuristic region-based methods detect text-based on edge or stroke information, i.e. strength, density or distribution. Sato et al. [4] apply 3 × 3 horizontal differential filter to the entire video frame with appropriate binary thresholding followed by size, fill factor and horizontal–vertical aspect ratio constraints. Anthimopoulos et al. [5] use Canny edge map followed by morphological operations and projection analysis. Kim et al. [6] instead of using an explicit edge map as an indicator of overlay text, they suggest the use of a transition map generated by the change of intensity and a modified saturation. A heuristic rule based on the different Local Binary Patterns (LBPs) is used for verification. Chen et al. [7] embed multiresolution and multiscale edge detection, adaptive searching, colour analysis, and affine rectification in a hierarchical framework for sign detection. Epshtein et al. [8] detect strokes in natural scene images and group them based on stroke width and geometrical constraints. These heuristic techniques proved to be very efficient and satisfactory robust for specific applications with high contrast characters and relatively smooth background. However, the fact that many parameters have to be estimated experimentally condemns them to data dependency and lack of generality.

DCT coefficients globally map the periodicity of intensity images and they have been widely used as texture features for heuristic texture-based methods [9–12]. Goto [13] employs Fisher's discriminant analysis to improve a DCT-based feature set and suggests the use of an unsupervised thresholding method for discriminating text and non-text regions. Moreover, DCT coefficients can be a quite efficient solution for jpeg and mpeg encoded images and videos. In this case, the pre-computed coefficients of 8 × 8 pixel block units are used. However, this block size is not a large enough area to sufficiently depict the periodical features of a text line and the computation of DCT for larger windows even by the fast DCT transform proves quite costly. In addition, these methods still use empirical thresholds on specific DCT-based features and therefore they lack adaptability.

Some hybrid methods have also been proposed. These methods usually consist of two stages. The first localizes the text with a fast heuristic technique while the second verifies the previous results eliminating some detected area as false alarms using machine learning. Machine learning classifiers have proved to be an appealing solution for many problems that cannot be defined in a strict mathematical manner. In [14], Chen et al. use a localization/verification scheme with the verification stage based on an SVM classifier trained on Constant Gradient Variance (CGV) features. Ye et al. [15] propose a coarse to fine algorithm that uses wavelets. The first stage applies thresholding on the wavelet energy to coarsely detect text, while the second identifies the coarse results using an SVM and a more sophisticated feature set that captures various wavelet characteristics. Jung et al. [16] apply as a first stage, a stroke filtering and they also verify the result using an SVM with normalized gray intensity and CGV features. Then, a text line refinement module follows, consisting of text boundary shrinking, combination and extension functions. Anthimopoulos et al. [17] propose a two-stage scheme that detects coarsely videotext based on edge density and then refines the result using an SVM and a new feature set based on the edge Local Binary Pattern (eLBP) operator which describes the spatial distribution of edges. Ye et al. [18] combine colour connected component analysis, texture classification by an SVM trained on wavelet histogram and OCR statistic features in a coarse-to-fine framework to discriminate texts from non-text patterns in natural scene images. Ji et al. [19] proposed a coarse to fine architecture which applies the LBP operator on the Haar wavelets. The first stage of the algorithm uses heuristic rules while the second one is based on an SVM classifier. Ekin [20] proposes a hardware-oriented artificial text detection algorithm that integrates a CC-based algorithm with a texture-based machine learning approach. Hybrid techniques combine the efficiency of heuristic methods with machine-learning accuracy and generalization. However, the inability of heuristic methods to handle cases with complex background and the need for a high recall rate from the first stage, often leads to a huge number of false alarms and forces the computationally expensive machine learning stage to scan nearly the whole image.

Several machine learning, texture-based approaches have been proposed for the detection of text areas with great success. These methods use directly machine learning classifiers to detect text. Jung [21] and Kim et al. [22] use the color and gray values of the pixels as input for a Neural Network (NN) and an SVM, respectively. Wolf et al. [23] use an SVM trained on differential and geometrical features. Lienhart et al. [24] used as features the complex values of the two-directional RGB gradient of the input image fed to a complex-valued NN. Li et al. [25] suggest the use of the mean, second-order (variance) and third-order central moments of the LH, HL, and HH component of the first three levels of each window to train a three-layer NN. The main shortcoming of the methods attributed to this category is the high computational complexity since a sliding window is required to scan the entire image with a typical step of three or four pixels, requiring thousands of calls to the SVM or NN classifier per image. To overcome the problem of complexity caused by SVM and NN, Chen et al. [26], based on Viola-Jones [27] method for face detection, use a cascade of AdaBoost strong classifiers for fast detection of scene text. Each of these AdaBoost classifiers are based on a number of weak classifiers trained on edges, histogram of intensities, gradient direction and intensity features. The AdaBoost cascade combines the generalization of a machine-learning method with a very fast prediction, giving satisfactory results in a very efficient way. However, the feature set needs to be reconsidered and adapted to the specific problem while a post-processing stage is required to refine the text localization. Motivated by the success of the AdaBoost algorithm for face and text detection we explored different classification algorithms, keeping in mind the need for very fast prediction and good generalization. Random Forests (RFs) proved to have all the desirable properties and combined with the proposed feature set and the adaptive post-processing step gave as superior results compared against the state-of-the-art.

3 Proposed method

In this work, we propose the use of an RF within a sliding window model for the discrimination of text areas in the first stage, and then apply a gradient-based algorithm to achieve separation and refined localization of the text lines (Fig. 2). This is actually a hybrid scheme combining an initial machine learning, texture-based technique with a heuristic, region-based refinement. The algorithm described above constitutes a fixed-scale detector so it is applied in multiple resolutions to detect text of any size. After processing all the resolutions needed, the final text line bounding boxes have been computed. Then, text lines are binarized and optionally segmented into words based on the distances between the resulting connected components. This segmentation is applied in cases where the text detection targets words instead of text lines.

The main contributions of this work is the choice of the classifier which provides efficiency and generalization capabilities, together with an improved, highly discriminative feature set that was designed particularly for reflecting the textual characteristics. Moreover, the use of a machine learning architecture for the first and most crucial stage, produces a generic and robust system for the detection of artificial and scene text in camera-based images and video frames. In our previous work [17], we proposed a hybrid system with a first heuristic edge-based stage followed by a second machine-learning stage for refining the initial results. This refinement stage was based on an SVM classifier and a feature set that describes the spatial distribution of luminance edges for different contrast levels. In the proposed work, the description of the edge spatial distribution is done in the RGB color space considering the valuable color information while the contrast levels adapt locally to each area of the image producing an actual parameter-free feature set. The use of the SVM in [17] forced us to invent a reduced version of the feature set in order to have an efficient system. This fact resulted in loss of information for the description of the textual texture. Contrary, in this work the use of an RF and its capability to deal efficiently with high dimensional feature spaces allowed us to use the whole proposed feature set instead of a reduced version so all available information is exploited for a better description of texture. Finally, the impressive efficiency of the RF gave us the capability to use it as the first and basic stage for scanning the whole image and detecting text instead of just refining the coarse results of a heuristic and unreliable stage.

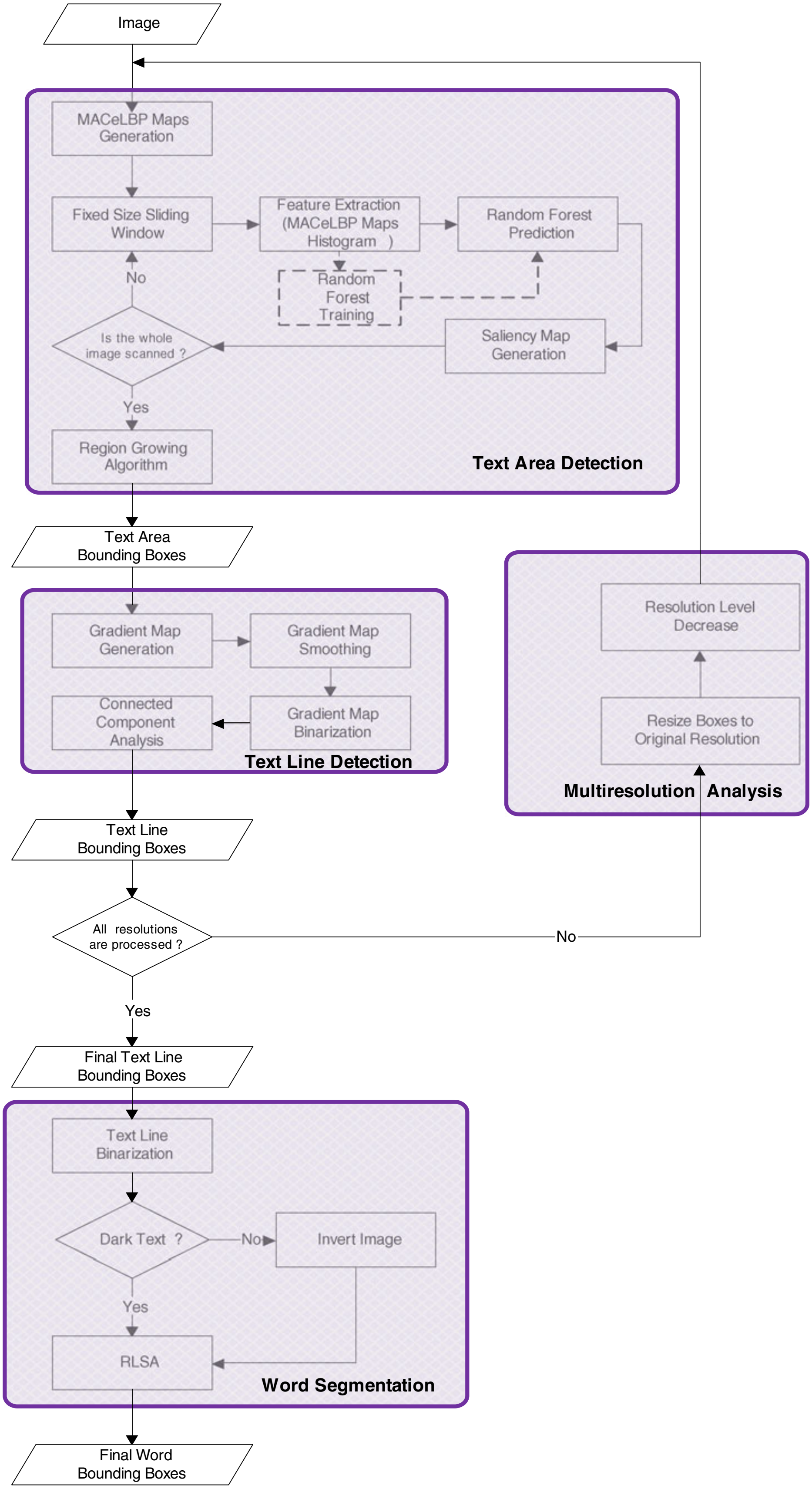
3.1 Text area detection

In this stage we assume that text areas constitute a distinct texture and we treat text detection as a texture segmentation problem. To do that, we need a fast and accurate classifier and a set of discriminative features which can also be computed rapidly.

3.1.1 Feature maps generation

The chosen feature set is based on the eLBP operator originally proposed in [17]. The eLBP is a modified LBP operator which actually describes the local edge patterns appeared in an image, for different levels of detail. LBP was originally introduced by Ojala et al. [28] as a non-parametric operator measuring the local contrast for efficient texture classification. The LBP operator consists of a 3 × 3 kernel where the centre pixel is used as a threshold. Then the eight binarized neighbours are multiplied by the corresponding binomial weight producing an integer in the range [0…0255] (Fig. 3). Each of the 256 different 8-bit words is considered to represent a unique texture pattern.

Fig. 2 Flowchart of the proposed text detection algorithm



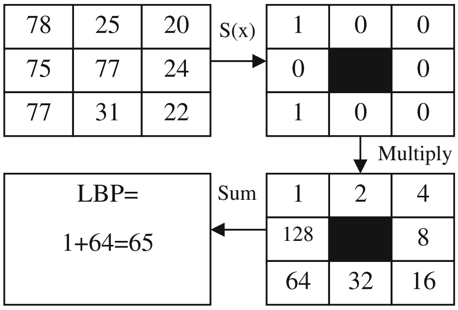


Fig. 3 Example of LBP computation

Formally, the decimal form of the resulting 8-bit word (LBP code) can be expressed as follows:

$$\mathrm{LBP}(x_c, y_c) = \sum_{n=0}^{7} S(i_n - i_c)2^n \tag{1}$$

where i_c corresponds to the grey value of the centre pixel (x_c, y_c), i_n ($n \in [0, 7]$) to the grey values of the eight neighbouring pixels, and function $S(x)$ is defined as:

$$S(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

When LBP is applied in a greyscale image, another 8-bit greyscale image is created in which each pixel value represents the texture pattern of the corresponding pixel in the original image. Thus, the 256 histogram values of an image region depict its texture structure. Although the original LBP operator has shown satisfactory performance for many kinds of texture classification, in the case of textual texture it cannot support a representation which results in adequate performance. To this end, we proposed the eLBP [17] which actually describes the spatial distribution of edges appeared in an image which constitutes the fundamental text characteristic. In eLBP, a neighbouring pixel is represented by 0 if it is close to the centre pixel or 1 if not (Fig. 4). In other words, a neighbouring pixel is assigned the value of 1 only if it constitutes an edge with respect to the centre pixel.

Formally, the eLBP operator is defined as:

$$\mathrm{eLBP}(x_c, y_c) = \sum_{n=0}^{7} S_e(i_n - i_c)2^n \tag{3}$$

where function $S_e(x)$ is defined by:

$$S_e(x) = \begin{cases} 1, & |x| \geq e \\ 0, & |x| < e \end{cases} \tag{4}$$

The value of e determines how sharp an intensity change should be in order to be considered as an edge. It has to be large enough for avoiding the arbitrary intensity variations caused by noise and small enough to detect all the deterministic intensity changes of texture. To solve this problem we propose the generation of multilevel eLBP edge histograms with several values for the threshold e, which will describe the edge distribution in different contrast levels. In the sequel, a detailed description of the computation of multiple thresholds is given.

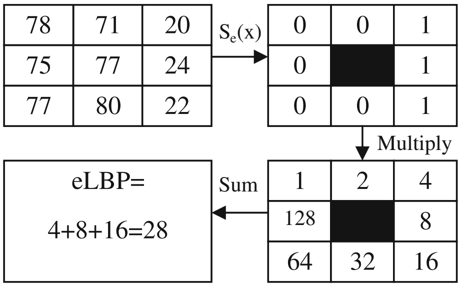


Fig. 4 Example of eLBP computation

From the Eqs. 3 and 4 we can see that the values of e will be the thresholds of the absolute differences between adjacent pixels which actually constitute the gradient of the image. We can safely assume that the probability density function (PDF) of an image's gradient, has a Laplacian distribution, hence the PDF of the absolute gradient values will be exponential. Figure 5 shows a typical PDF of an image gradient. As it can be seen, the probability falls for high distance values. The threshold e at Eq. 4 would actually binarize this distribution to distinguish edge from non-edge pixels and then operator (3) will combine the neighbouring edge values to generate edge patterns. The optimal set of thresholds for the multilevel edge description will have to cluster the gradient's PDF in clusters with equal probability. To this end, we fit the distribution of the image gradient values to the exponential distribution:

$$PDF_{\exp}(x) = \lambda \times e^{-\lambda \times x} \tag{5}$$

where λ is the rate parameter of the distribution and x is the random variable. To achieve that, we can calculate the mean value of image gradient and set it equal to λ^{-1}. An example of PDF fitting can be seen in Fig. 5.

The quantile function (inverse cumulative distribution function) of $\mathrm{PDF}_{\exp}$ is denoted as:

$$F_{\exp}(l) = -\lambda^{-1} \times \ln(1 - l) \tag{6}$$

where $0 \leq l < 1$. The quantile function returns a boundary below which random values from the given distribution would fall, with a probability equal to l. Therefore, to cluster the distribution in equal probability clusters we use as thresholds the values of $F_{\exp}$ for equally spaced values of l in [0,1) as shown in Eq. 7

$$l = i/L + 1 \tag{7}$$

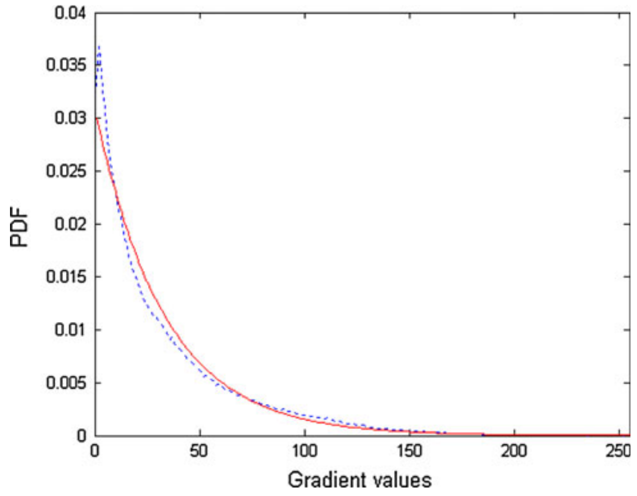


Fig. 5 Typical distribution of image gradient values (*dotted*), fitted exponential distribution (*solid*)

where $i = 1\ldots L$ and L denotes the number of different levels. In that way, the selected threshold values will be concentrated close to zero where the PDF shows higher values.

Contrary to our previous work [17], instead of calculating the mean of the whole gradient image to find the thresholds of different levels, we use the mean value of a 9 × 9 area around each corresponding pixel so the multilevel binarization of the gradients also becomes locally adaptive. In other words, this PDF fitting will be done for every single pixel of the image considering the gradient's local mean value, re-establishing the threshold set and giving to the feature set the ability to adapt to the image areas with different contrast profile. Besides, instead of computing the gradient from the intensity image we use the RGB color image, thus the difference between any two adjacent pixels is defined as the maximum difference appeared in the three RGB color channels. In that way, we expose edges between different colors even if they correspond to the same intensity value.

The number of required levels of contrast to capture the whole edge information is estimated experimentally; the selection of eight levels does work satisfactorily for most applications. Increasing the number of contrast levels would increase the extracted information for the spatial distribution of edges as well as the number of features. However, after relative experimentation it was shown that eight levels give adequate information even for the most detailed and distorted edge distributions. For easy cases that guarantee high contrast between text and background and minimum image distortion, fewer levels can be used in order to accelerate the whole method. Considering the need for a fast machine learning stage and the large number of features for every sliding window (i.e. 256 × 8 = 2,048) we have to be very efficient on the way that feature values are computed. First, four absolute directional gradient maps are generated, namely vertical, horizontal, diagonal and anti-diagonal (Fig. 6). Then, four maps with the Gradient Local Mean are computed by smoothing the initial maps with a Gaussian mask (Fig. 7). Let us call GM_i, and GLMM_i the gradient maps and the corresponding local mean maps, where $i = 1, 2, 3, 4$, denote the four different directions (horizontal, diagonal, vertical, anti-diagonal). Each of the gradient maps is going to be adaptively binarized and create four edge maps for each of the L levels:

$$\mathrm{EM}_i^l(x,y) = \begin{cases} 1, \mathrm{GM}_i(x,y) > e_i^l(x,y) \\ 0, \mathrm{GM}_i(x,y) \leq e_i^l(x,y) \end{cases} \tag{8}$$

where $i = 1, 2, 3, 4$, $l = 1\ldots L$ and

$$e_i^l(x,y) = -\mathrm{GLMM}_i(x,y) \times \ln(1-l) \tag{9}$$

Then, the four edge maps of each level will supply the binarized values of $S(x)$ in Eq. 3 and L different Multilevel Adaptive Color eLBP (MACeLBP) maps will be created. The new MACeLBP is defined as:

$$\begin{aligned}\mathrm{MACeLBP}^l(x,y) &= \mathrm{EM}_1^l(x,y) \times 2^0 + \mathrm{EM}_2^l(x,y) \times 2^1 \\ &+ \mathrm{EM}_3^l(x,y) \times 2^2 + \mathrm{EM}_4^l(x,y) \times 2^3 \\ &+ \mathrm{EM}_1^l(x,y-1) \times 2^4 + \mathrm{EM}_2^l(x-1,y-1) \times 2^5 \\ &+ \mathrm{EM}_3^l(x-1,y) \times 2^6 + \mathrm{EM}_4^l(x-1,y+1) \times 2^7\end{aligned} \tag{10}$$

with $l = 1\ldots L$.

Figure 8 presents the adaptive thresholded maps (EM_i^l) for the fifth level ($l = 5$) while Fig. 9 displays the corresponding $\mathrm{MACeLBP}^5$ map. From Fig. 8 we can notice the high edge density in almost every area of the image, caused by the adaptive binarization. This means that the generated features will not rely on the edge density or gradient magnitude but in edge spatial distribution to distinguish text from background, capturing the characteristics of even minimum contrast characters. The only information regarding the strength of edges comes actually from the differences between the several contrast levels. The reader may also notice from Fig. 9 that raw MACeLBP values do not have any obvious meaning as they are just integers assigned to specific edge patterns. However, the histogram values of an image area give a very informative description of the different edge patterns frequency.

3.1.2 Random forest

RF are ensemble classifiers proposed by Breiman [29]. An RF is a combination of decision trees such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. To classify a new input vector, each tree gives a classification, and we say the tree "votes" for that class.

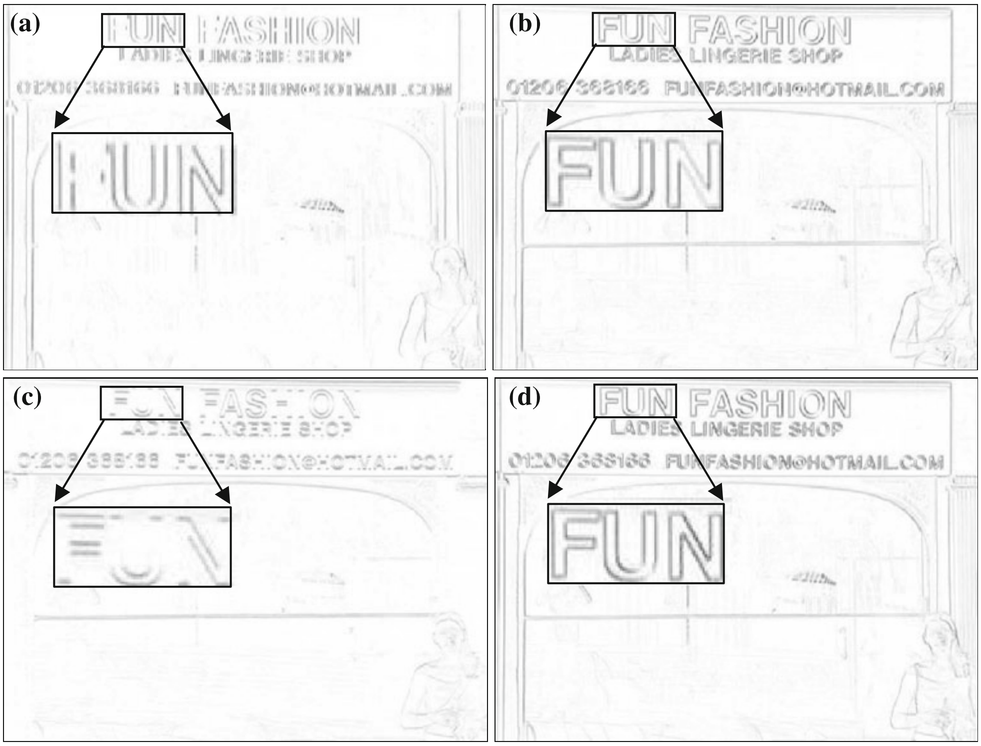


Fig. 6 The absolute directional gradient maps (*inverted*). **a** Horizontally, **b** diagonally, **c** vertically, **d** anti-diagonally

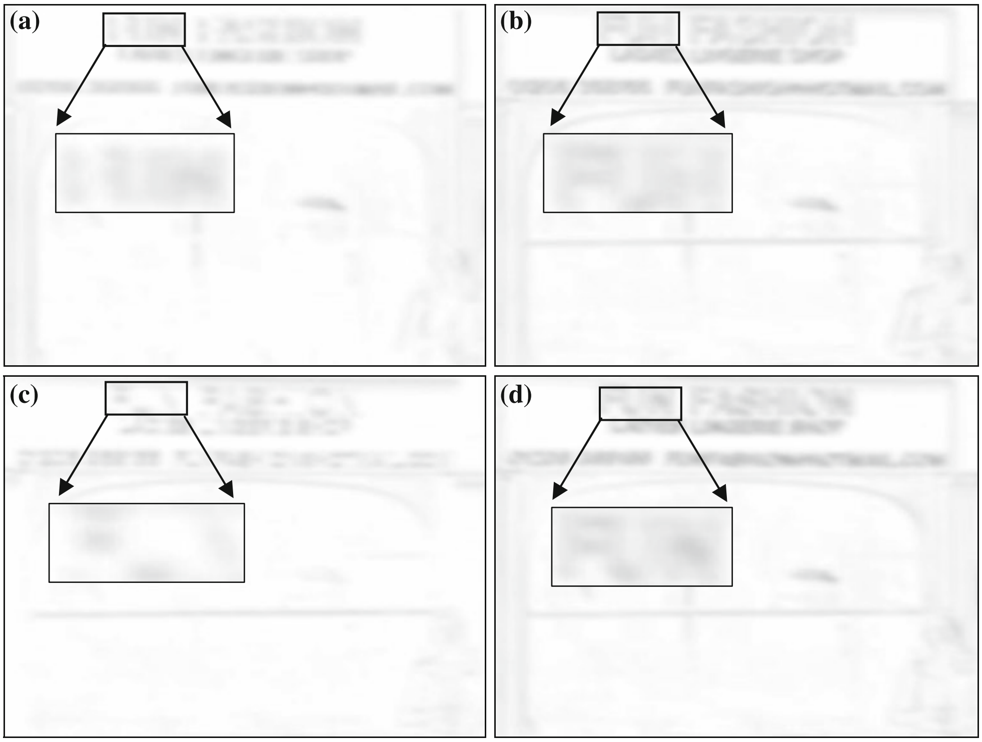


Fig. 7 The gradient local mean maps (*inverted*). **a** Horizontally, **b** diagonally, **c** vertically, **d** anti-diagonally

The forest chooses the classification having the most votes over all the trees in the forest or returns the average of tree responses as a non-binary response. RFs have been used in numerous machine learning applications with a great success and present several appealing characteristics. The classification performance of RFs is reported to be comparable to SVM, NNs and Boosting [30, 31] while at the same time RFs are faster in training and predicting, fully

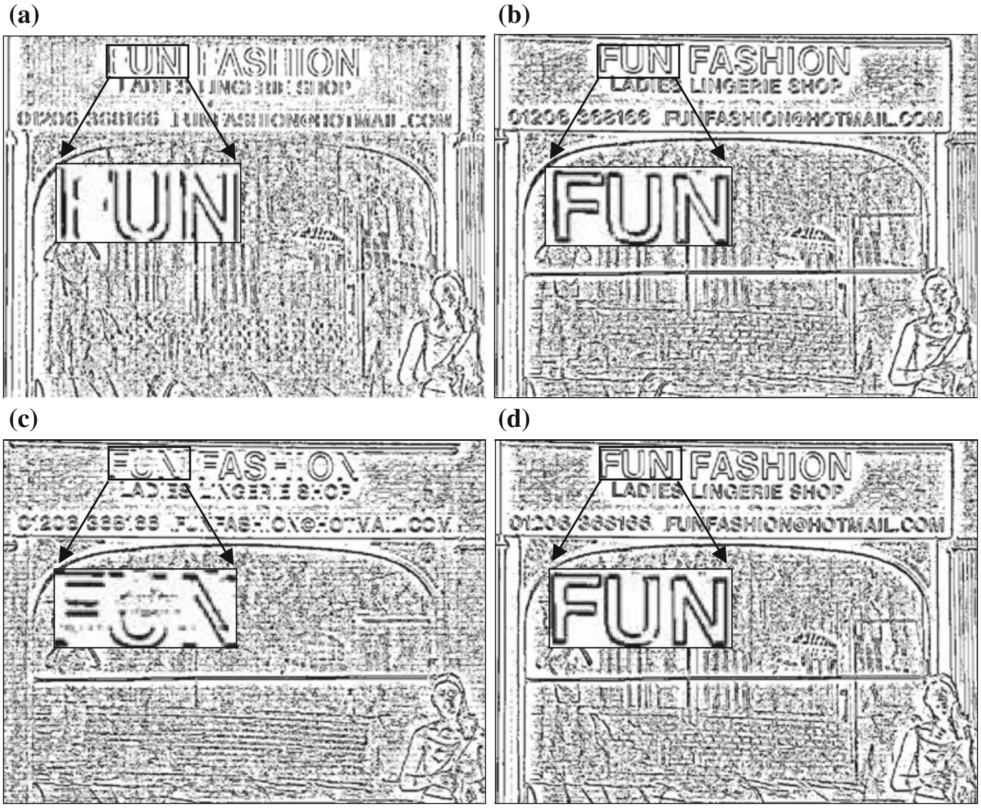


Fig. 8 The adaptively threshold gradient maps for level $l = 5$ (*inverted*). **a** Horizontally, **b** diagonally, **c** vertically **d** anti-diagonally

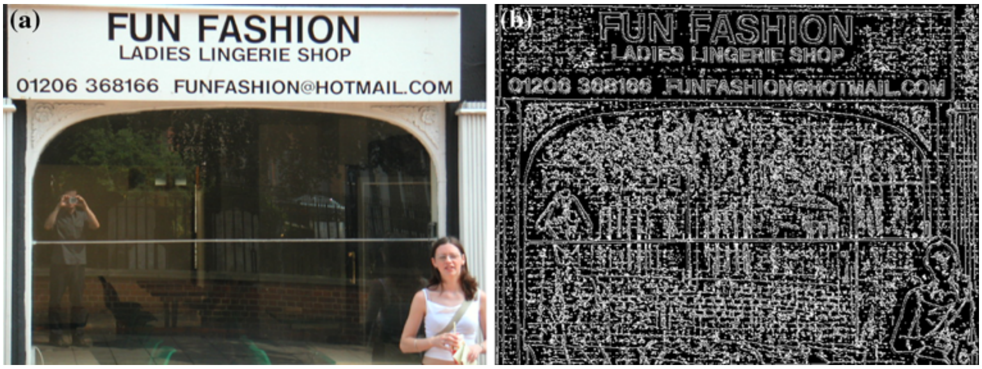


Fig. 9 **a** The original image, **b** the MACeLBP^l map for level $l = 5$

parallelizable and easily implemented. Two very important advantages of RFs against Boosting are their abilities not to overfit the data and deal robustly with noise. Another useful ability is that they can handle very large numbers of input variables and select the best features by estimating the importance of each variable. The multilevel nature of the MACeLBP features increase the dimension of the feature set to $256{\times}L$, where L is the number of levels, making the choice of RF ideal for the specific application. In our previous work [17] the use of an SVM classifier forced as to invent a reduced version of the proposed feature set to have an efficient system. However, this reduction resulted in loss of information for the description of textual texture.

3.1.3 Saliency map generation: region growing algorithm

After the creation of the MACeLBP maps, calculation of the histograms for each contrast level will be applied over sliding windows. The window's height is equal to the height of the shortest text that is expected to be detected but width equals twice the height assuming that a word contains at least two characters. The histogram values for each window position will not be computed from scratch. While the window slides within the MACeLBP maps, the majority of histogram values remains the same, so the algorithm needs just to add the values of the newly inserted pixels and remove the values of the pixels that where excluded from the window after its last slide. In that way, feature generation becomes much more efficient and its complexity achieves independence from the sliding step.

The histogram values of all levels in each window position will constitute the features that will be fed to the RF classifier for classifying the window as text or non-text. In order to obtain all possible information, the classifier returns a confidence value instead of a binary decision. This confidence value is actually the average response of random trees and expresses the probability of having text. These text probabilities are added to the saliency map and then a Gaussian smoothing follows. After the saliency map generation a region growing algorithm is applied. Two thresholds *th1* and *th2* (with *th1* > *th2*) are used to define whether an area of the map belongs to text. All the pixels of the map with value over *th1* are considered to belong to the text and therefore they are used as seeds. Also, if the value of a pixel is below *th1* but over *th2* and has a neighboring pixel already classified as text it is also considered as a text pixel. The values of *th1* and *th2* are experimentally estimated to 2/3 and 1/2, respectively, which means that a text region must contain at least one pixel classified as text with a confidence over 66.7% (usually in the center of text) but all pixels have to be classified as text with at least 50% confidence. A connected component analysis follows, producing one output bounding box for every text area. Figure 10a presents the smoothed saliency map while Fig. 10b shows the corresponding bounding boxes. It can be seen that the upper box contains two different text lines which actually have different font size. The next step of the algorithm will have to separate these text lines and provide a more accurate localization.

3.2 Text line localization

Although the sliding window model has shown great discrimination capabilities between text and non-text areas, it appears to meet difficulties in accurate localization and segmentation of text lines as shown in Fig. 10. More accurate ways for text line and word localization should rely on a region-based algorithm, i.e. it should utilize the image gradient, edges or connected components. At this step, bounding boxes may contain several text lines with different fonts, colors and size, so a gradient-based method seems more appropriate than relying on color homogeneity. Based on these observations, we firstly compute the Sobel gradient magnitude for every detected box from the machine learning stage. Since this gradient map may also contain some non-text objects we multiply it by the saliency map to suppress their gradient values. Afterwards, we smooth the gradient map with a Gaussian mask, binarize it using Otsu thresholding [32] and generate one bounding box for every created connected component that satisfies the size restrictions, namely its height should lie within the specific range of the fixed scale detector. Figure 11 shows this procedure for the upper box of Fig. 10b. The gradient magnitude used here is a sum of the vertical and horizontal absolute Sobel gradients with weights 0.8 and 0.2, respectively, that emphasize the vertical one. The Gaussian mask we use for the smoothing has a small height equal to three (as used in our experiments) and width equal to the box height.

Figure 12 displays an example of the text line localization algorithm applied in an image with skewed text lines. Moreover, this example explains the reason for the multiplication with the saliency map which was produced by the sliding window model. If someone looks carefully the initial Sobel Gradient map, he would see some vertical strokes that do not belong to any characters. These non-text gradient values are suppressed after the multiplication with the saliency map, making the gradient based text line segmentation more accurate.

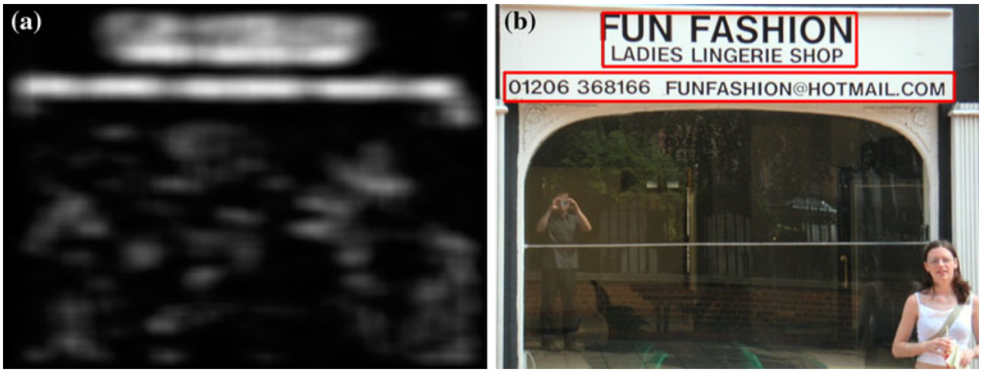


Fig. 10 **a** Saliency map of the second multiresolution level, **b** the corresponding bounding boxes

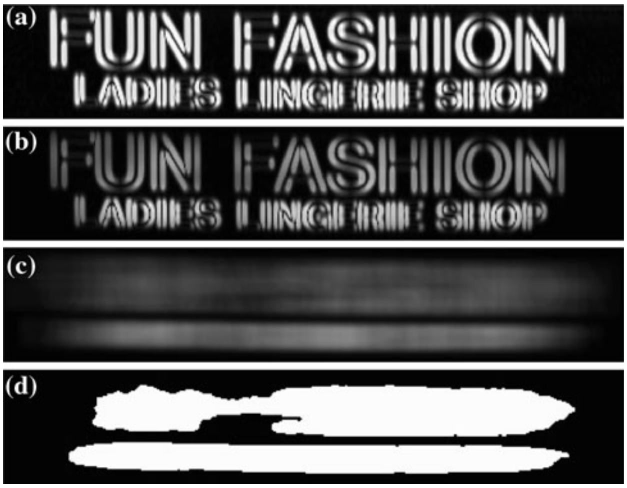


Fig. 11 Text line localization. **a** Sobel gradient, **b** gradient after multiplying with saliency map, **c** gradient after Gaussian smoothing, **d** gradient after Otsu thresholding

3.3 Multiresolution analysis

The machine learning, sliding window model described above constitutes a fixed-scale text line detector, namely it is capable of detecting text in a narrow range of font heights since the properties of textual texture depends on the text size. In order to make the system scale independent we adopt a multiresolution approach. The whole text line detector is applied to the image in different resolutions and finally the results are combined in a map with the original resolution. Narrowing the range of the fixed scale detector results in smaller text inter-class variance hence provide better classification accuracy. On the other hand, a narrow range would also lead to many levels for the multiresolution approach and thus will increase the complexity of the algorithm. Eventually, the choice of the range has to satisfy the tradeoff between the performance of the fixed-scale detector and the system's efficiency. Figure 13 presents an example of detecting characters with different sizes in the appropriate resolutions.

3.4 Word segmentation

Many publications related to text detection consider the text line as text unit while others aim at detecting words. The algorithm described until this section outputs one bounding box for every text line so the stage we are going to present here is optional and intents to segment text lines producing word bounding boxes. The decision about including or not the word segmentation module in the text detection system depends on the needs of the specific application or the annotation strategy of the dataset used for experimentation. This stage is based on connected components instead of edges since we can safely assume that each text line contain characters of the same color. Before identifying words in text lines we have to binarize the image and make sure that the black pixels correspond to text. This is an essential step for word segmentation but it is also needed for text recognition which is the actual goal of a text extraction system.

3.4.1 Image binarization: invert text detection

For the binarization of the images, Otsu [32] thresholding was chosen after relative experimentation. Otsu method applies the optimum separation of the two dominant colors in an image based on the minimization of their intra-class variance. Moreover, this separation is symmetrical contrary to other binarization methods meaning that it will have equivalent performance for both normal and inverse text. To classify between normal or inverse text we first apply a connected component analysis. The numbers of white (WCC) and black (BCC) connected components are

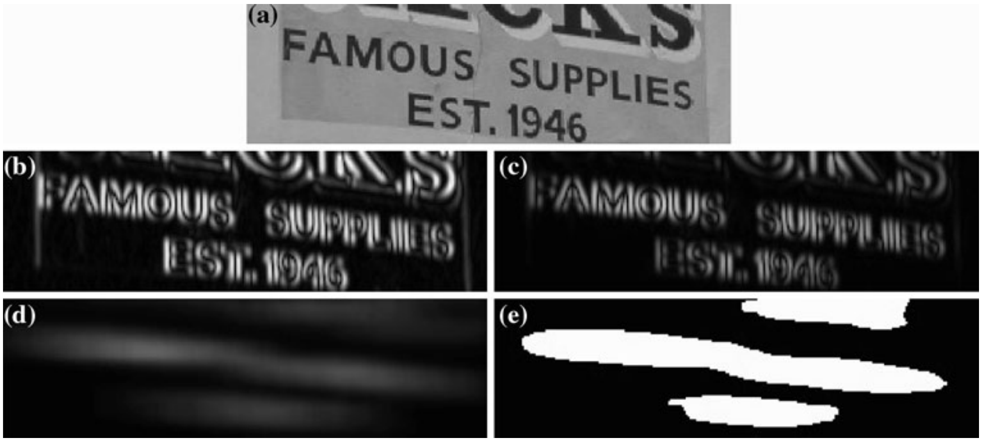


Fig. 12 Text line localization, a skew example. **a** Original image, **b** Sobel gradient, **c** gradient after multiplying with saliency map, **d** gradient after Gaussian smoothing, **e** gradient after Otsu thresholding

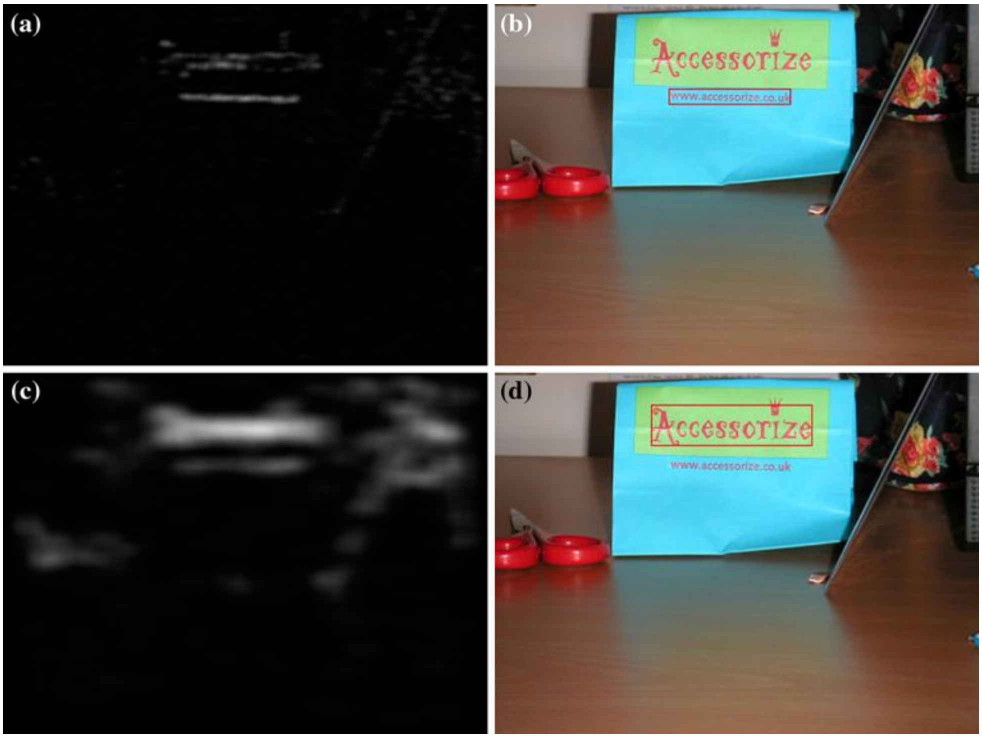


Fig. 13 Multiresolution detection. **a** Saliency map of 1st scale, **b** result of 1st scale, **c** saliency map of 4th scale, **d** result of 4th scale

counted, discarding components with height less than eight pixels or less than one-third of the box height. If |WCC–BCC| >1 then the color that corresponds to the largest number of connected components is regarded as text color. Else if the distance between WCC and BCC equals 0 or 1, the condition for the inversion is based on the pixel values of the borders of the bounding boxes. If the majority of border pixels are black then text is considered inverse. Finally, a color inversion is applied to every inverse text image so the output of this step consists of binarized normal text.

3.4.2 *Run length smoothing algorithm*

The input of this step is a binarized image with black for text and white for background pixels. For the separation of the different words in a text line we have to vertically cut the bounding box wherever the distance between two adjacent characters is relatively large. In order to define "relatively large", we calculate the mean value (M) and the standard deviation (SD) of all horizontal distances between black pixels of different connected components. After that, we apply a Run Length Smoothing Algorithm (RLSA) that connects any pair of black pixels having horizontal distance less than $M - 0.2 \times \mathrm{SD}$. For any black connected component produced, a new bounding box is created which is actually the final word bounding box. Figure 14 presents the steps of word segmentation for the upper text line of Fig. 11.

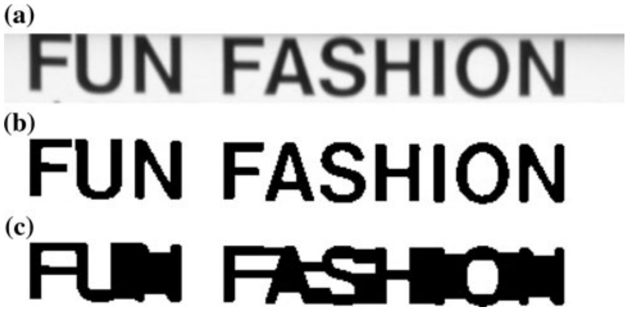


Fig. 14 Word segmentation. **a** Grayscale image, **b** binarized image, **c** RLSA result

4 Experimental results and discussion

The proposed methodology has been extensively tested using a concise evaluation methodology which comprises multiple datasets, a variety of alternate features, different evaluation measures and a comparison with various state-of-the-art methodologies. All the experiments were conducted using one 2.4 GHz core of an Intel Q6600 CPU. To evaluate the system's performance in different stages and scopes we used five datasets:

- Dataset 1: It comprises 3,944 text and 5,701 non-text image samples with size 24 × 12 and was created to test the classification accuracy and efficiency of different combinations of features and classifiers for the text area detection as described in Sect. 3.1.

- Dataset 2: It consists of 3,048 normal and 1,827 inverse text images taken from the ground truth or the results of several text detection methods and it was used for the experiments regarding the normal/inverse text classification, described in Sect. 3.4.1.

The rest three datasets were used for evaluating the final results of the proposed text detection system in different kinds of data.

- Dataset 3: The third dataset is the public dataset of ICDAR 2003 Robust Reading competition. This is a very difficult dataset presenting all the challenges of the natural scene text such as variant illumination and complex background. The whole set contains 509 images from which 258 belong to the training set and the rest 251 to the test set.
- Dataset 4: This is the first set of our previous work [17] which contains 214 video frames from athletic events with 2,963 text occurrences. The text appearing in this set is mainly artificial text from video captions.
- Dataset 5: The final dataset we used for our experiments is the second set of [17] which includes 172 frames from news and advertisements with a total of 672 artificial text occurrences.

The experiments on Dataset 1 compare the discrimination abilities of five different feature sets combined with SVM and RF classifiers. The feature sets we used except from the proposed MACeLBP include the Reduced eLBP of [17] and three of the most popular feature sets in literature: DCT coefficients, Haar wavelets and gradient values for our tests, we used the raw values of LH, HL and HH components of the first three levels of Haar decomposition since they have shown better performance than any other wavelet-based feature set. Also, the first coefficient of DCT transform is omitted since it is proportional to the intensity mean and does not contain any frequency information. The gradient feature set corresponds to the gradient Sobel magnitude values. For the experiments we used an SVM with an RBF kernel and an RF consisting of 20 trees each of them chooses randomly 200 out of 256 × 8 = 2,048 features. Table 1 presents the related classification results using cross-validation with 10-folds while Table 2 compares SVM and RF regarding the speed of prediction. The related recall and precision measures are defined as:

$$\mathrm{recall}_{\mathrm{text}} = \frac{\text{correctly classified text samples}}{\text{overall text samples}} \tag{11}$$

$$\mathrm{Precision}_{\mathrm{text}} = \frac{\text{correctly classified text samples}}{\text{overall samples classified as text}} \tag{12}$$

Table 1 Classification results of *Dataset 1* for different features and classifiers

Features	Feature dimension	$\mathrm{Recall}_{\mathrm{text}}$	$\mathrm{Precision}_{\mathrm{text}}$	F_{text}
RF				
MACeLBP	2,048	96.6	97.4	97
Reduced eLBP	256	93.5	94.6	94.1
DCT	287	94.2	91.1	92.7
Haar	279	91.4	90	90.1
Gradient	288	90.5	88.2	89.3
SVM				
Reduced eLBP	256	98	98	98
MACeLBP	2,048	96	98.2	97.1
DCT	287	94.2	96.2	95.2
Haar	279	93.1	95.7	94.4
Gradient	288	80.1	92.3	86.3

Table 2 Comparing SVM and RF in terms of prediction speed

Predictions/sec	SVM	RF
MACeLBP	40	~150,000
Reduced eLBP	160	~500,000
DCT	180	~500,000
Haar	385	~500,000
Gradient	200	~500,000

It is obvious that RF is by far faster than SVM which is practically non-applicable since scanning a 1,280 × 960 image with a sliding step of two pixels requires over 300,000 predictions that is, it requires at least 30 min only for the classification part. On the other hand, the results of RF are at least comparable with the corresponding results of SVM while in some cases are even better. An interesting observation about the results is that the best performance is achieved by the reduced eLBP feature set fed to SVM classifier as proposed in our previous work [17]. The MACeLBP set presented slightly worst performance combined with SVM, despite carrying more information, and this is probably because of the dimension expanding that made feature space sparser. However, the difference in terms of performance between SVM and RF best results is negligible compared to the corresponding computational cost. It is noted that in [17] the use of SVM was possible since the sliding window model was used only for refining small image parts produced by the first coarse detection stage.

The experiments upon the Dataset 2 proved the successful performance of the normal/inverse text classification since the proposed method scored 97% in terms of accuracy. The rest 3% for which the particular technique failed corresponds mainly to cases where text detection also failed, producing bounding boxes that cut text lines horizontally or contain too much non-text content.

For testing the performance of the proposed system on scene text detection we used Dataset 3 (ICDAR 2003). The size of the sliding window was set to 24×12 pixels while the sliding step is 2. The fixed-scale detector was trained to detect text in height range 12–18 and the multiresolution uses ten levels with a scale factor of 2/3. For the results we used the recall and precision measures as defined in [33] to compare with the competition results:

$$p = \frac{\sum_{r_e \in E} m(r_e, T)}{|E|} \tag{13}$$

$$r = \frac{\sum_{r_t \in T} m(r_t, E)}{|T|} \tag{14}$$

in which

$$m(r, R) = \max m_a(r, r') | r' \in R \tag{15}$$

$$m_a(r1, r2) = \frac{2 \times a \times (r1 \cap r2)}{a(r1) + a(r2)} \tag{16}$$

$a(r)$ is the area of rectangle r while $|E|$ is the number of detected boxes and $|T|$ the number of ground truth bounding boxes.

As overall measure, the harmonic mean is used:

$$f = \frac{2 \times p \times r}{p + r} \tag{17}$$

Table 3 presents the performance evaluation results of the proposed algorithm as well as the average time, compared to the Robust Reading competitions of 2003 and 2005 and some more recent works. The results prove the robustness of the algorithm in a very challenging dataset. However, the annotation strategy and the used evaluation protocol need to be reconsidered. In some cases there is no typical and objective way that word bounding boxes are defined by the annotator. For example, the symbol "@", in Fig. 10b, is not considered as a character while other symbols like "&" are considered as characters. In some images word bounding boxes may contain dashes but in other images dashes are considered to separate words so they are excluded from bounding boxes. There are also several wrong or missing text boxes from the ground truth which produce misleading results. Another issue that affects the systems result is the existence of isolated characters. Our system just like many text detection systems aims at detecting words with at least two characters since one character has no periodicity and thus does not constitute a texture. Besides, the majority of arbitrary connected components in an image resemble the shape of at least one character. The omission of isolated characters is mainly the reason for the relative low recall rate of the proposed system.

Table 3 Comparative results for *Dataset 3* using the ICDAR2003 evaluation protocol and average processing time

Method	p	r	f	T (sec)
Proposed	0.82	0.61	0.70	4.2
Ji [19]	0.59	0.79	0.68	–
Epshtein [8]	0.73	0.60	0.66	0.94
Robust reading competition 2005				
Hinnerk Becker	0.62	0.67	0.62	14.4
Alex Chen	0.60	0.60	0.58	0.35
Qiang Zhu	0.33	0.40	0.33	1.6
Jisoo Kim	0.22	0.28	0.22	2.2
Nobuo Ezaki	0.18	0.36	0.22	2.8
Robust reading competition 2003				
Ashida	0.55	0.46	0.50	8.7
HWDavid	0.44	0.46	0.45	0.3
Wolf	0.30	0.44	0.35	17
Todoran	0.19	0.18	0.18	0.3

Furthermore, the evaluation rates of the whole Robust Reading test set as described in [33] come from computing the recall, precision and f rates for every image and then averaging over all images. This means that every image will contribute the same to the final result despite the number of containing boxes which is not intended in a box-based evaluation protocol. Moreover, if the detector returns no boxes for an image, the precision is wrongly considered to be zero while actually it cannot be defined.

Another shortcoming is that although the ICDAR2003 measures intent to estimate the proportion of correctly detected word bounding boxes, they cannot deal with the problem of splits or merges between boxes. This means that if a resulting box covers an entire text line, the protocol will map it against only one of the containing words giving unfairly low results. Wolf et al. [34] proposed the creation of match score matrices with the overlap between every possible pair of blocks to consider the possible splits and merges besides one-to-one matching. However, to match two ground truth boxes with one resulting box, the total overlap threshold (as described in Wolf et al. paper) has to be very low ($\sim$40%). This will have as a result accepting as correct, a box with size even higher than the double size of the ground truth box. Table 4 presents the results of the proposed method using Wolf's evaluation protocol, compared to the corresponding results of Robust Reading 2003 competition as reported in [34].

Finally Dataset 4 and Dataset 5 were used to test the system's performance on video frames. Dataset 5 contains only artificial text while Dataset 4 comprises some occurrences of scene text, too. For these sets, the proposed system was applied without the final word segmentation

Table 4 Comparative results for *Dataset 3* using Wolf evaluation protocol [34]

Method	$\text{Precision}_{\text{Wolf}}$	$\text{Recall}_{\text{Wolf}}$	f_{Wolf}
Proposed	79.3	61.4	69.2
Robust Reading Competition 2003			
Ashida	41.7	55.3	47.5
HWDavid	46.6	39.6	42.8
Wolf	44.9	19.4	27.1
Todoran	17.9	14.3	15.9

Table 5 Comparative results for *Dataset 4* and *Dataset 5*

%	Method	$\text{Recall}_{\text{ecn}}$	$\text{Precision}_{\text{ecn}}$	F_{ecn}
Dataset 4	[22]	66.9	66.7	66.8
	[14]	65.4	75.6	70.1
	[25]	81.1	70.5	75.4
	[17]	83.9	79	81.4
	Proposed	84.1	79.8	81.9
Dataset 5	[22]	63.3	69.2	66.1
	[14]	68.2	71.1	69.6
	[25]	80.6	71.5	75.8
	[17]	82.7	83.5	83
	Proposed	83.4	84.7	84

Table 6 Average processing time for *Dataset 4* and *Dataset 5* (sec/per frame)

Method	Average processing time per frame
[22]	8
[14]	3.35
[25]	1.5
[17]	2
Proposed	1.6

stage since the annotation for the specific dataset was done in a text line level. Table 5 shows the corresponding results while Table 6 presents the average processing time per frame. For the comparison with the results of this dataset, we used the recall and precision rates as defined in [17].

The evaluation protocol in [17] was based on the recall and precision of the area coverage, normalised by the estimated number of characters for every box (see Eqs. 18, 19). The number of characters in a bounding box was approximated by the ratio width/height of the box, assuming that this ratio is invariable for every character, the spaces between different words in a text line are proportional to its height and each textline contains characters of the same size.

$$\text{Recall}_{\text{ecn}} = \frac{\sum_{i=1}^{N} \frac{|\text{GDI}_i|}{\text{hg}_i^2}}{\sum_{i=1}^{N} \frac{|\text{GB}_i|}{\text{hg}_i^2}} \tag{18}$$

$$\text{Precision}_{\text{ecn}} = \frac{\sum_{i=1}^{M} \frac{|\text{DGI}_i|}{\text{hd}_i^2}}{\sum_{i=1}^{M} \frac{|\text{DB}_i|}{\text{hd}_i^2}} \tag{19}$$

where GB_i is the ground truth bounding box number i and hg_i is its height, while DB_i is the detected bounding box number i and hd_i is its height. N is the number of ground truth bounding boxes and M is the number of detected bounding boxes and GDI, DGI are the corresponding intersections:

$$\text{GDI}_i = \text{GB}_i \cap \left(\bigcup_{i=1}^{M} \text{DB}_i \right) \tag{20}$$

$$\text{DGI}_i = \text{DB}_i \cap \left(\bigcup_{i=1}^{N} \text{GB}_i \right) \tag{21}$$

Table 5 together with the previous results proves the capability of the proposed system to detect artificial and scene text in video frames or camera-based images. The datasets used for the evaluation contain a great variety of fonts in different sizes and colors and present all kinds of challenges from both artificial and scene text. The experiments we conducted confirmed our initial assumption that the discrimination between text and complex background constitutes a common prominent challenge for every kind of text detection application.

5 Conclusion

In this paper we presented a method for text detection in images. The system consists of a machine learning stage which is based on an RF classifier and a very discriminating proposed feature set, followed by a gradient-based text line localization step. Finally a connected component based algorithm segments the detected text line in words. The whole system works in a multiresolution manner providing detection of characters in a wide range of sizes. The main contributions of this work is the highly discriminating feature set based on a new texture operator, combined with the accuracy and efficiency of the RF classifier. Experimental results have been produced using a concise evaluation methodology and show the superior performance achieved on the detection of both artificial and scene text embedded in very complex backgrounds. As future work we could adjust the MACeLBP–RF model for different detection purposes since many kinds of objects or textures are described by their spatial edge distribution.

References

1. Lienhart R, Effelsberg W (2000) Automatic text segmentation and text recognition for video indexing. ACM/Springer Multiméd Sys 8:69–81
2. Sobottka K, Bunke H, Kronenberg H (1999) Identification of text on colored book and journal covers. International conference on document analysis and recognition, pp 57–63
3. Wang K, Kangas JA (2003) Character location in scene images from digital camera. Pattern Recognit 36(10):2287–2299
4. Sato T, Kanade T, Hughes E, and Smith M (1998) Video ocr for digital news archives, IEEE workshop on content-based access of image and video databases, pp 52–60
5. Anthimopoulos M, Gatos B, Pratikakis I (2007) Multiresolution text detection in video frames. International conference on computer vision theory and applications, pp 161–166
6. Kim W, Kim C (2009) A new approach for overlay text detection and extraction from complex video scene. IEEE Trans Image Process 18(2):401–411
7. Chen X, Yang J, Zhang J, Waibel A (2004) Automatic detection and recognition of signs from natural scenes. IEEE Trans Image Process 13(1):87–99
8. Epshtein B, Ofek E, Wexler Y (2010) Detecting text in natural scenes with stroke width transforms, IEEE conference on computer vision and pattern recognition, San Francisco
9. Zhong Y, Zhang H, Jain AK (2000) Automatic caption localization in compressed video. IEEE Trans Pattern Anal Machine Intell 22(4):385–392
10. Crandall D, Antani S, Kasturi R (2003) Extraction of special effects caption text events from digital video. Int J Document Anal Recognit 5(2–3):138–157
11. Lim Y.K, Choi S.H, and Lee S.W (2000) Text extraction in mpeg compressed video for content-based indexing. International conference on pattern recognition, pp 409–412
12. Gargi U, Crandall D.J, Antani S, Gandhi T, Keener R, Kasturi R (1999) A system for automatic text detection in video. International conference on document analysis and recognition, pp 29–32
13. Goto H (2008) Redefining the DCT-based feature for scene text detection: Analysis and comparison of spatial frequency-based features. Int J Document Anal Recognit 11(1):1–8
14. Chen D, Odobez J-M, Thiran J-P (2004) A localization/verification scheme for finding text in images and videos based on contrast independent features and machine learning methods. Image Commun 19(3):205–217
15. Ye Q, Huang Q, Gao W, Zhao D (2005) Fast and robust text detection in images and video frames. Image Vision Comput 23(6):565–576
16. Jung C, Liu Q, Kim J (2009) A stroke filter and its application to text localization. Pattern Recogn Lett 30(2):114–122
17. Anthimopoulos M, Gatos B, Pratikakis I (2010) A two-stage scheme for text detection in video images. Image Vision Comput 28(9):1413–1426
18. Ye Q, Jiao J, Huang J, Yu H (2007) Text detection and restoration in natural scene images. J Vis Commun Image Represent 18(6):504–513
19. Ji R, Xu P, Yao H, Zhang Z, Sun X, Liu T (2008) Directional correlation analysis of local Haar binary pattern for text detection. IEEE International Conference on Multimedia & Expo, pp 885–888
20. A. Ekin (2006) Information based overlaid text detection by classifier fusion. IEEE international conference on acoustics, speech and signal processing, pp II-753–II-756
21. Jung K (2001) Neural network-based text location in color images. Pattern Recogn Lett 22(14):1503–1515
22. Kim KI, Jung K, Park SH, Kim HJ (2001) Support vector machine-based text detection in digital video. Pattern Recogn 34(2):527–529
23. Wolf C and Jolion J-M (2004) Model Based Text Detection in Images and Videos: a Learning Approach. Technical Report LIRIS-RR-2004-13 Laboratoire d'Informatique en Images et Systemes d'Information, INSA de Lyon, France
24. Lienhart R, Wernicke A (2002) Localizing and segmenting text in images and videos. IEEE Trans Circuits and Systems for Video Technol 12(4):256–268
25. Li H, Doermann D, Kia O (2000) Automatic Text Detection and Tracking in Digital Video. IEEE Trans Image Process 9(1):147–156
26. Chen X.R, Yuille A.L (2004) Detecting and reading text in natural scenes. IEEE computer society conference on computer vision and pattern recognition, pp 366–373
27. Viola PA, Jones MJ (2004) Robust real-time face detection. Int J Comp Vision 57(2):137–154
28. Ojala T, Pietikainen M, Harwood D (1996) A comparative study of texture measures with classification based on feature distributions. Pattern Recogn 29(1):51–59
29. Breiman L (2001) Random forests. Machine Learn 45(1):5–32
30. Tang Y, Krasse S, He Y, Yang W, Alperovitch D (2008) Support vector machines and random forests modeling for spam senders behavior analysis. GLOBECOM, pp 2174–2178
31. Bosch A, Zisserman A, Munoz X (2007) Image classification using random forests and ferns, 11th IEEE international conference on computer vision, pp 1–8
32. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE transactions on systems. Man Cybern 9(1): 62–66
33. Lucas S, Panaretos A, Sosa L, Tang A, Wong S, Young R (2003) ICDAR 2003 robust reading competitions, ICDAR, pp 682–687
34. Wolf C, Jolion J (2006) Object count/area graphs for the evaluation of object detection and segmentation algorithms. Int J Doc Anal Recognit 8(4):280–296