Neurocomputing 168 (2015) 23–34

Contents lists available at ScienceDirect

Neurocomputing

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

A robust hybrid method for text detection in natural scenes by learning-based partial differential equations

Zhenyu Zhao [a], Cong Fang [b], Zhouchen Lin [b,c,*], Yi Wu [a]

[a] Department of Mathematics, School of Science, National University of Defense Technology, PR China
[b] Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, PR China
[c] Cooperative Medianet Innovation Center, Shanghai Jiaotong University, PR China

ARTICLE INFO

Article history:
Received 8 April 2015
Received in revised form
22 May 2015
Accepted 10 June 2015
Communicated by Luming Zhang
Available online 20 June 2015

Keywords:
Text detection
Natural scenes
Hybrid method
Learning-based PDEs

ABSTRACT

Learning-based partial differential equations (PDEs), which combine fundamental differential invariants into a non-linear regressor, have been successfully applied to several computer vision tasks. In this paper, we present a robust hybrid method that uses learning-based PDEs for detecting texts from natural scene images. Our method consists of both top-down and bottom-up processing, which are loosely coupled. We first use learning-based PDEs to produce a text confidence map. Text region candidates are then detected from the map by local binarization and connected component clustering. In each text region candidate, character candidates are detected based on their color similarity and then grouped into text candidates by simple rules. Finally, we adopt a two-level classification scheme to remove the non-text candidates. Our method has a flexible structure, where the latter part can be replaced with any connected component based methods to further improve the detection accuracy. Experimental results on public benchmark databases, ICDAR and SVT, demonstrate the superiority and robustness of our hybrid approach.

1. Introduction

Text detection and recognition in natural scene images have received more and more attention in recent years [1–4]. This is because text often provides critical information for understanding the high-level semantics of multimedia content, such as street view data [5,6]. Moreover, the demand of a growing number of applications on mobile devices has brought great interest in this problem. Text detection from natural scene images is very challenging due to the complexity of background, uneven lighting, blurring, degradation, distortion, and the diversity of text patterns.

There have been a lot of methods for scene text detection, which can be roughly divided into three categories: sliding window based methods [7–9], connected component (CC) based methods [5,10,11], and hybrid methods [12]. Sliding window based methods search for possible texts in multi-scale windows in an image and then classify them into positives using a lot of texture features. However, they are often computationally expensive when a large number of windows, with various sizes, need to be checked and complex classification methods are used. CC-based methods firstly extract character candidates as connected components using some low-level features, e.g., color similarity and spatial layout. Then the character candidates are grouped into words after eliminating the wrong ones by connected components analysis (CCA). The hybrid method [12] creates a text region detector to estimate the probabilities of text position at different scales and extract character candidates (connected components) by local binarization. The CC-based and the hybrid methods are more popular than the sliding window based ones because they can achieve a high precision once the candidate characters are correctly detected and grouped. However, such a condition is not often met: the low-level operations are usually unreliable and sensitive to noise, which makes it difficult to extract the right character candidates. The large number of wrong character candidates can cause many difficulties in the post-processing, such as grouping and classification.

Recently, Liu et al. [13] have proposed a framework that learns partial differential equations (PDEs) from training image pairs, which has been successfully applied to several computer vision and image processing problems. It can handle some mid-and-high-level tasks that the traditional PDE-based methods cannot. In [13] they apply learning-based PDEs to object detection, color2-gray, and demosaicking. In [14], they use an adaptive (learning-based) PDEs system for saliency detection. However, these methods [13,14] may not handle text detection well. This is because text

* Corresponding author at: Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, PR China.
E-mail addresses: dwightzzy@gmail.com (Z. Zhao), tjufangcong19911231@gmail.com (C. Fang), zlin@pku.edu.cn (Z. Lin), wuyi_work@sina.com (Y. Wu).

http://dx.doi.org/10.1016/j.neucom.2015.06.019
0925-2312/

Fig. 1. Examples of the detected text region candidates (in green boxes) by learning-based PDEs in natural scene images (*Images in this paper are best viewed on screen!*).

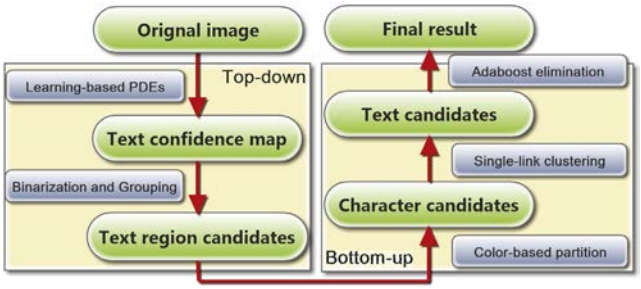

Fig. 2. Pipeline of the proposed approach.

is a man-made object and its interpretation strongly depends on human perception. The complexity of backgrounds, flexible text styles, and variation of text contents make text detection more challenging than previous tasks.

In most cases, texts cover only a small area of the scene image (Fig. 1). So it will be beneficial for post-processing, like character candidate extraction, if we could narrow down the candidates of text region and eliminate annoying backgrounds. Although we cannot expect that the learned PDEs can produce an exactly binary text mask map, because the solution of the learned PDEs should be a more or less smooth function, learning-based PDEs can give us a relatively high quality confidence map as a good reference. So we use learning-based PDEs to design a text region detector. It is much faster than sliding window based methods because its complexity is only $O(N)$, where N is the number of pixels in the image. Some examples containing the detected region candidates are shown in Fig. 1. To make our method complete and comparable to others, we further propose a simple method for detecting texts in the region candidates.

In summary, we propose a new robust hybrid method using learning-based PDEs. It incorporates both top-down scheme and bottom-up scheme to extract texts in natural scene images. Fig. 2 shows the flow chart of our system and Fig. 3 gives an example. A PDEs system is first learnt off-line with L_1-norm regularization on the training images. In the top-down scheme, given a test image as the initial condition we solve the learnt PDEs to produce a high quality text confidence map (see Fig. 3(b)). Then we apply a local binarization algorithm (Niblack [15]) to the confidence map to extract text region candidates (see Fig. 3(c)). In the bottom-up scheme, we present a simple connected component based method and apply it to each region candidate to determine accurate text locations. We firstly perform mean shift algorithm and binarization (OTSU [16]) to extract character candidates (see Fig. 3(d)). Then we group these components to text lines simply based on their color and size (see Fig. 3(e)). Next we adopt a two-level classification scheme (character candidates classifier and text candidates classifier) to eliminate the non-text candidates. Then we obtain the final result (Fig. 3(f)). Our system is evaluated on several benchmark databases and has achieved higher F-measures than other methods. Note that *the parameters and classifiers are only trained on the ICDAR 2011 database* [17] as only this database provides the required training information. But the proposed approach still yields higher precisions and recalls on the SVT databases [5,6] than other state-of-the-art methods. We summarize the contributions of this paper as follows:

- We propose a new hybrid method for text detection from natural scene images. Unlike previous methods, our method consists of loosely coupled top-down and bottom-up schemes, where the latter part can be replaced by any connect component based methods.
- We apply learning-based PDEs for computing a high quality text confidence map, upon which good text region candidates can be easily chosen. Unlike sliding window based methods, the complexity of learning-based PDEs for text candidate proposal is only $O(N)$, where N is the number of pixels. So our learning-based PDEs are much faster. To our best knowledge, this is the first work that applies PDEs to text detection.
- We conduct extensive experiments on benchmark databases to prove the superiority of our method over the state-of-the-art ones in detection accuracy. Note that unlike previous approaches, after computing the text confidence map, all the procedures are very simple. *The performance could be further improved if more sophisticated and ad hoc treatments are involved.*

The rest of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 describes the top-down scheme and Section 4 describes the bottom-up scheme. We discuss the relationship between our method and some related work in Section 5. Section 6 presents experiments that compare the proposed method and the state-of-the-art ones on several public databases. We conclude our paper in Section 7.

2. Related work

Most sliding window based methods first search for possible texts in multi-scale windows and then estimate the text existence probability by using classifiers. Zhong et al. [1] adopt image transformations, such as discrete cosine transform and wavelet decomposition, to extract features. They remove the non-text regions by thresholding the filter responses. Kim et al. [7] extract texture features from all local windows in every layer of image pyramid, which enables the method to detect texts at variable scales. They use a fast mode seeking process named continuously adaptive mean-shift (CAMSHIFT) to search for the text positions and an SVM classifier to generate text probability maps. Li et al. [18] use first and second order moments of wavelet decomposition responses as local region features and a neural network classifier to filter the negative candidates. To speed up text detection, Chen and Yuille [8] propose a fast text detector using a cascade AdaBoost classifier, whose weak learners are selected from a feature pool containing gray-level, gradient, and edge features. After classification the windows are further grouped with morphological operations [19], conditional random fields [6] or graph based methods [20]. The advantage of these methods lies in the simple and adaptive training-detection architecture. Yet that a large number of window candidates need to be classified results in their expensive computational cost.

Unlike sliding window based methods, CC-based methods first extract character candidates from images by connected component analysis and then group the character candidates into words. Additional classifier may be used to eliminate false positives. CC-based methods have become the focus of several recent work thanks to their low computational cost. In addition, the located text components can be directly used for recognition. Based on the color uniformity of characters in a text string, Yi and Tian [21] propose a color-based partition scheme, which applies weighted *k*-means clustering in the RGB space to separate text and background pixels. In [22] Shivakumara et al. filter the image in the frequency domain, by using the Fourier–Laplacian transform, and then apply *k*-means clustering to identify candidate components. Recently, two methods, Maximally Stable Extremal Regions (MSER) [23,24] and Stroke Width Transform (SWT) [5,25,26] have been widely used because of their effectiveness of extracting character/component candidates. Yin et al. [24] use a pruning algorithm to select appropriate MSERs as character candidates and hybrid features to validate the candidates, achieving state-of-the-art performance on the ICDAR 2011 database [17]. Using SWT, Yao et al. [25] follow Epshtein et al.'s work [5] in pixel-level filtering and grouping. Then, after running heuristic filtering, two classifiers were trained and applied to remove the outliers in components and text lines. Huang et al. [26] develop a novel Stroke Feature Transform (SFT) filter and two Text Covariance Descriptors (TCDs) for text detection and get a significant performance improvement. In implementations of CCs, syntactic pattern recognition methods are often used to analyze the spatial and feature consensus and to define text regions.

The most related work to ours is the recently proposed hybrid method by Pan et al. [12]. They create a text region detector which computes the text confidence values of sub-windows by using the

Table 1
Fundamental differential invariants up to the second order, where tr is the trace operator and ∇f and $\mathbf{H}_f$ are the gradient and the Hessian matrix of function f, respectively.

i	$\mathrm{inv}_i(u,v)$
0, 1, 2	$1, v, u$
3, 4	$\|\nabla v\|^2 = v_x^2 + v_y^2$, $\|\nabla u\|^2 = u_x^2 + u_y^2$
5	$(\nabla v)^T \nabla u = v_x u_x + v_y u_y$
6, 7	$\mathrm{tr}(\mathbf{H}_v) = v_{xx} + v_{yy}$, $\mathrm{tr}(\mathbf{H}_u) = u_{xx} + u_{yy}$
8	$(\nabla v)^T \mathbf{H}_v \nabla v = v_x^2 v_{xx} + 2v_x v_y v_{xy} + v_y^2 v_{yy}$
9	$(\nabla v)^T \mathbf{H}_u \nabla v = v_x^2 u_{xx} + 2v_x v_y u_{xy} + v_y^2 u_{yy}$
10	$(\nabla v)^T \mathbf{H}_v \nabla u = v_x u_x v_{xx} + (v_x u_y + u_x v_y) v_{xy} + v_y u_y v_{yy}$
11	$(\nabla v)^T \mathbf{H}_u \nabla u = v_x u_x u_{xx} + (v_x u_y + u_x v_y) u_{xy} + v_y u_y u_{yy}$
12	$(\nabla u)^T \mathbf{H}_v \nabla u = u_x^2 v_{xx} + 2u_x u_y v_{xy} + u_y^2 v_{yy}$
13	$(\nabla u)^T \mathbf{H}_u \nabla u = u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}$
14	$\mathrm{tr}(\mathbf{H}_v^2) = v_{xx}^2 + 2v_{xy}^2 + v_{yy}^2$
15	$\mathrm{tr}(\mathbf{H}_v \mathbf{H}_u) = v_{xx} u_{xx} + 2v_{xy} u_{xy} + v_{yy} u_{yy}$
16	$\mathrm{tr}(\mathbf{H}_u^2) = u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2$

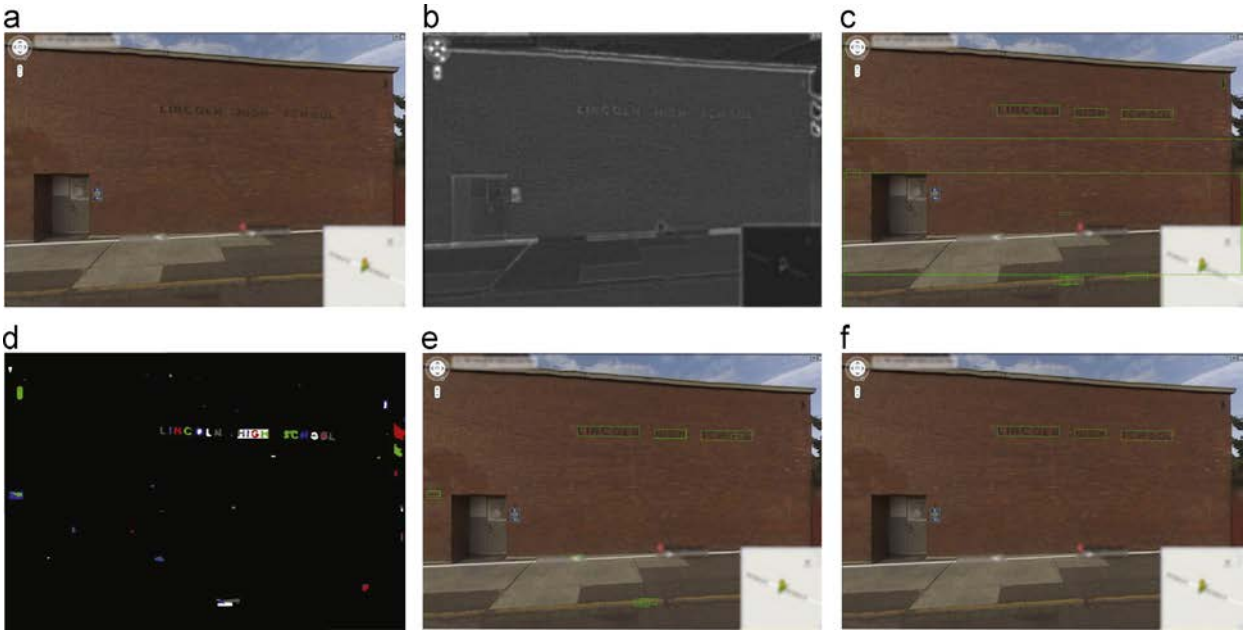

Fig. 3. Text detection process. (a) The original image. (b) Text confidence map. (c) Text region candidates (in green boxes). (d) Character candidates (in different colors). (e) Text candidates (in green boxes). (f) Final result (in green boxes). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

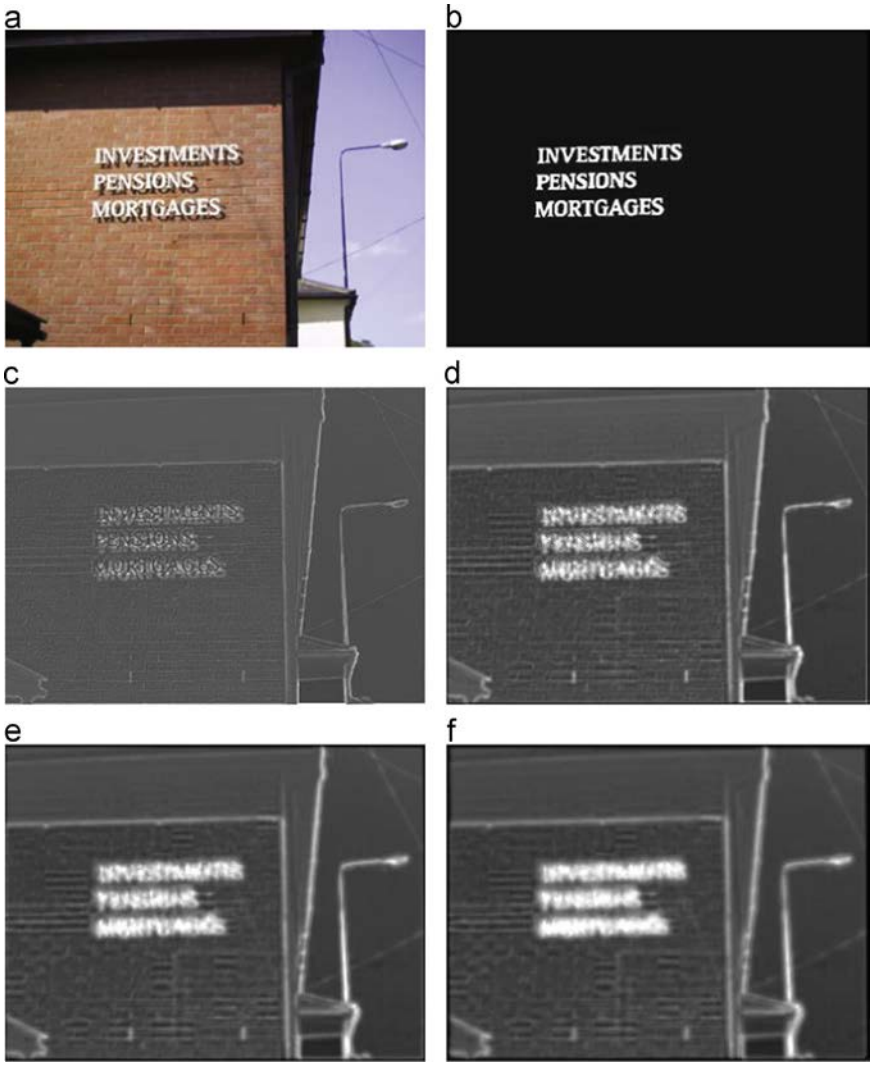

Fig. 4. Evolution of PDEs. (a) and (b) is an input/output training image pair. (c)–(e) show the evolutionary results of the learned PDEs at time $n=1$, 3, and 6, respectively. (f) is the text confidence map output by the learned PDEs.

Histogram of Oriented Gradients (HOG) feature and then extract connected components by local binarization. After that they use a conditional random field model to eliminate the non-character components and group text components into text lines/words with an energy minimization model. Our method computes the text confidence value of each pixel by learning-based PDEs and then group the pixels into text region candidates. It removes lots of backgrounds and hence makes the character candidates extraction much easier. After detecting text region candidates, any CC-based methods can be used. So our method has a flexible structure.

3. Top-down: text confidence map computation and text region candidates extraction

As mentioned before, the proposed method has both the top-down scheme and the bottom-up scheme. We introduce the top-down scheme in this section and leave the bottom-up scheme in next section. We first introduce learning-based PDEs and then apply them to compute the text confidence map. Then we adopt a local thresholding method for text region candidates extraction.

3.1. Text confidence map computation

In this subsection, we first give a brief overview on learning-based PDEs. Then we propose to use the L_1 norm regularization for the coefficients of PDEs and explain the algorithm for learning-based PDEs. At last, we introduce the numerical implementation for solving the PDEs to compute the text confidence map.

3.1.1. Learning-based PDEs

Learning-based PDEs have been successfully applied to several computer vision and image processing problems, such as denoising, deblurring, demosaicking, and saliency detection [13,14]. It is assumed that the evolution of the image u is guided by an indicator function v, which collects large scale information. It is grounded on the translational and rotational invariance of computer vision and image processing problems. Namely, when the input image is translated or rotated, the output image is translated or rotated at the same amount. It can be proven that the governing equations are functions of fundamental differential invariants (Table 1), which form "bases" of all differential invariants that are invariant with respect to translation and rotation [13].

Assuming linear combination of the fundamental differential invariants, the formulation of PDEs is as follows:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \sum_{i=0}^{16} a_i(t)\mathrm{inv}_i(u,v) = 0, & (x,y,t) \in Q, \\ u(x,y,t) = 0, & (x,y,t) \in \Gamma, \\ u(x,y,0) = f_u, & (x,y) \in \Omega, \\ \dfrac{\partial v}{\partial t} - \sum_{i=0}^{16} b_i(t)\mathrm{inv}_i(u,v) = 0, & (x,y,t) \in Q, \\ v(x,y,t) = 0, & (x,y,t) \in \Gamma, \\ v(x,y,0) = f_v, & (x,y) \in \Omega, \end{cases} \tag{1}$$

where f_u and f_v are the initial functions of u and v, respectively, which are determined by the input image I (Fig. 4a), $\Omega \subset \mathbb{R}^2$ is the (rectangular) region occupied by the image,[2] T is the temporal span of evolution which can be normalized as 1, $Q = \Omega \times [0,T]$, $\Gamma = \partial\Omega \times [0,T]$, and $\partial\Omega$ is the boundary of Ω. So learning the PDEs reduces to determining the linear combination coefficients $\mathbf{a} = \{a_i(t) \mid i = 0,\ldots,16\}$ and $\mathbf{b} = \{b_i(t) \mid i = 0,\ldots,16\}$ among the fundamental differential invariants. That the coefficients are functions of time t only and independent of spatial variables is a consequence of the translational and rotational invariance of the PDEs [13].

The optimal coefficients should minimize the difference between the output of PDEs, when the initial function is the input training image (see Fig. 4(a)), and the ground truth (output training image, see Fig. 4(b)) [13]. To this end, one may prepare a number of input/output training image pairs $((I_m, O_m),\ i = 1,\ldots,M)$, where I_m is the input training image and O_m is the output training image. This results in a PDEs constrained optimal control problem:

$$\min_{\mathbf{a},\mathbf{b}} E(\mathbf{a},\mathbf{b}) = \frac{1}{2}\sum_{m=1}^{M}\int_{\Omega}(O_m - u_m(x,y,T))^2\,\mathrm{d}\Omega + \lambda_1\sum_{i=0}^{16}\int_0^T a_i^2(t)\,\mathrm{d}t + \lambda_2\sum_{i=0}^{16}\int_0^T b_i^2(t)\,\mathrm{d}t, \tag{2}$$

where u_m, $\mathbf{a}$, and $\mathbf{b}$ satisfy Eq. (1) and $u_m(x,y,T)$ is the solution of PDEs (1) at $t=T$ when the input image is I_m (see Fig. 4(f)).

Model (2)–(1) is for grayscale images. When detecting texts, the input is usually a color image (see Fig. 4(a)) and the expected output is a binary image (the mask image at text position, see Fig. 4(b)). So we extend the learning-based PDEs framework to accommodate three indicate functions $\mathbf{v} = [v_1, v_2, v_3]^T$, each accounting for one channel of the color image and using one channel of the input image as its initial condition. u is simply initialized as an all-one function. Accordingly, there are four evolutionary PDEs, for u, v_1, v_2, and v_3, respectively. And the number of translationally and rotationally invariant fundamental differential invariants up to second order becomes 69. They are $\{1, f_r, (\nabla f_r)^T\nabla f_s, (\nabla f_r)^T\mathbf{H}_{f_m}\nabla f_s, \mathrm{tr}(\mathbf{H}_{f_r}), \mathrm{tr}(\mathbf{H}_{f_r}\mathbf{H}_{f_s})\}$ where $f_r, f_s, f_m \in \{u, v_1, v_2, v_3\}$. They can also be conveniently referred to as $\mathrm{inv}_i(u,\mathbf{v}), i = 0,\ldots,68$. Because of the significantly increased number of the fundamental differential invariants, the L_1-norm regularization is more suitable for the coefficients to encourage sparsity, i.e., using as few fundamental differential invariants as possible. Summing up, the color-image version of model (2)–(1) is

$$\min_{\mathbf{a},\mathbf{B}} E(\mathbf{a},\mathbf{B}) = \frac{1}{2}\sum_{m=1}^{M}\int_{\Omega}(O_m - u_m(x,y,T))^2\,\mathrm{d}\Omega + \lambda_1\sum_{i=0}^{68}\int_0^T |a_i(t)|\,\mathrm{d}t + \lambda_2\sum_{j=1}^{3}\sum_{i=0}^{68}\int_0^T |B_{i,j}(t)|\,\mathrm{d}t, \tag{3}$$

$$s.t.\ \begin{cases} \dfrac{\partial u_m}{\partial t} - \sum_{i=0}^{68} a_i(t)\ \mathrm{inv}_i(u_m,\mathbf{v}_m) = 0, & (x,y,t) \in Q, \\ u_m(x,y,t) = 0, & (x,y,t) \in \Gamma, \\ u_m(x,y,0) = \mathbf{1}, & (x,y) \in \Omega, \\ \dfrac{\partial v_{m,j}}{\partial t} - \sum_{i=0}^{68} B_{i,j}(t)\ \mathrm{inv}_i(u_m,\mathbf{v}_m) = 0, & (x,y,t) \in Q, \\ v_{m,j}(x,y,t) = 0, & (x,y,t) \in \Gamma, \\ v_{m,j}(x,y,0) = I_{m,j}, & (x,y) \in \Omega, \end{cases} \tag{4}$$

where $\mathbf{v}_m = [v_{m,1}, v_{m,2}, v_{m,3}]$, $\mathbf{a} = \{a_i(t) \mid i = 0,\ldots,68\}$ and $\mathbf{B} = \{B_{i,j}(t) \mid i = 0,\ldots,68, j = 1,2,3\}$.

3.1.2. Algorithm for solving the coefficients

Liu and Lin et al. [13] propose a gradient descent method to solve for the optimal coefficients $a_i(t)$ and $b_i(t)$ in (1), where the "gradient" is actually the Gâteaux derivatives [27] of the objective functional E with respect to the coefficient functions. The gradient descent method is time-consuming and only fits for smooth regularizations on the coefficient functions, such as the squared L_2 norm in (2).

Zhao et al. [28] propose a new algorithm by minimizing the difference between the expected outputs O_m (Fig. 4(b)) and the actual outputs $(u_m(x,y,t))$ of the PDEs at time t. We adopt this algorithm to solve the optimal control problem (3)–(4).

We first discretize the temporal variable t with a step size Δt and denote $t_i = i\cdot\Delta t$, $i = 0,\ldots,N$. In the sequel, for brevity we use u_m^n or $u_m(t_n)$ instead of $u_m(x,y,t_n)$ if no ambiguity can occur. Other notations, such as $\mathbf{v}_m^n$, $\mathbf{a}^n$, and $\mathbf{B}^n$ are understood similarly.

Forward scheme is used to approximate the governing equations in (4). Namely,

$$\begin{cases} u_m^{n+1} = u_m^n + \Delta t\ (\mathbf{a}^n)^T\cdot\mathbf{inv}(u_m^n,\mathbf{v}_m^n),\ n \ge 0, \\ \mathbf{v}_m^n = \mathbf{v}_m^{n-1} + \Delta t\ \left(\mathbf{B}^{n-1}\right)^T\cdot\mathbf{inv}(u_m^{n-1},\mathbf{v}_m^{n-1}),\ n \ge 1, \end{cases} \tag{5}$$

where $(\mathbf{a}^n)^T\cdot\mathbf{inv}(u_m^n,\mathbf{v}_m^n) \triangleq \sum_{i=0}^{68} a_i^n\ \mathrm{inv}_i(u_m^n,\mathbf{v}_m^n)$ and

$$\left(\mathbf{B}^{n-1}\right)^T\cdot\mathbf{inv}(u_m^{n-1},\mathbf{v}_m^{n-1}) \triangleq \left[\sum_{i=0}^{68} B_{i,1}^{n-1}\ \mathrm{inv}_i(u_m^{n-1},\mathbf{v}_m^{n-1}), \sum_{i=0}^{68} B_{i,2}^{n-1}\ \mathrm{inv}_i(u_m^{n-1},\mathbf{v}_m^{n-1}), \sum_{i=0}^{68} B_{i,3}^{n-1}\ \mathrm{inv}_i(u_m^{n-1},\mathbf{v}_m^{n-1})\right]^T.$$

Following Zhao et al. [28]'s scheme, we relax to minimize the difference between the expected outputs O_m and the actual outputs u_m^{n+1} of the PDEs in time order. Then the problem reduces to

$$\min_{\mathbf{a}^n,\mathbf{B}^{n-1}} L^{n+1}\left(\mathbf{a}^n,\mathbf{B}^{n-1}\right) = \frac{1}{2}\sum_{m=1}^{M}\int_{\Omega}(O_m - u_m^{n+1})^2\,\mathrm{d}\Omega + \lambda_1\|\mathbf{a}^n\|_1 + \lambda_2\|\mathbf{B}^{n-1}\|_1, \tag{6}$$

at each time t_n, where $\|\mathbf{a}^n\|_1 = \sum_{i=0}^{68}|a_i^n|$ is the sum of the absolute values of its components and $\|\mathbf{B}^{n-1}\|_1 = \sum_{j=1}^{3}\sum_{i=0}^{68}|B_{i,j}^{n-1}|$.

For $n=0$ (time t_0), to solve for $\mathbf{a}^0$, by using the first equation in (5) problem (6) reduces to minimize

$$L(\mathbf{a}^0) = \frac{1}{2}\sum_{m=1}^{M}\int_{\Omega}\left[O_m - u_m^0 - \Delta t\ (\mathbf{a}^0)^T\cdot\mathbf{inv}(u_m^0,\mathbf{v}_m^0)\right]^2\mathrm{d}\Omega + \lambda_1\|\mathbf{a}^0\|_1. \tag{7}$$

[2] The images are padded with zeros of several pixels width around them such that the Dirichlet boundary conditions $u(x,y,t) = 0, v(x,y,t) = 0, (x,y,t) \in \Gamma$, are naturally fulfilled.

When using binomial expansion and interchanging integration and summation, the first term of (7) can be rewritten as follows:

$$\begin{aligned}&\frac{(\Delta t)^2}{2}\sum_{m=1}^{M}\int_{\Omega}(\mathbf{a}^0)^T\cdot\mathbf{inv}(u_m^0,\mathbf{v}_m^0)\cdot\mathbf{inv}^T(u_m^0,\mathbf{v}_m^0)\cdot\mathbf{a}^0\,\mathrm{d}\Omega\\&-\Delta t\sum_{m=1}^{M}\int_{\Omega}(O_m-u_m^0)\mathbf{inv}^T(u_m^0,\mathbf{v}_m^0)\cdot\mathbf{a}^0\mathrm{d}\Omega\\&+\frac{1}{2}\sum_{m=1}^{M}\int_{\Omega}(O_m-u_m^0)^2\,\mathrm{d}\Omega\\&=\frac{(\Delta t)^2}{2}(\mathbf{a}^0)^T\cdot\sum_{m=1}^{M}\int_{\Omega}\mathbf{inv}(u_m^0,\mathbf{v}_m^0)\cdot\mathbf{inv}^T(u_m^0,\mathbf{v}_m^0)\,\mathrm{d}\Omega\cdot\mathbf{a}^0\\&-\Delta t\sum_{m=1}^{M}\int_{\Omega}(O_m-u_m^0)\mathbf{inv}^T(u_m^0,\mathbf{v}_m^0)\,\mathrm{d}\Omega\cdot\mathbf{a}^0\\&+\frac{1}{2}\sum_{m=1}^{M}\int_{\Omega}(O_m-u_m^0)^2\,\mathrm{d}\Omega.\end{aligned}$$

We can see that it is a quadratic programming problem with L_1-norm regularization for $\mathbf{a}^0$. Many efficient algorithms that guarantee globally optimal solutions have been proposed, such as Gradient Projection (GP) [29], Homotopy [30], Iterative Shrinkage-Thresholding (IST) [31], Accelerated Proximal Gradient (APG) [32], and Alternating Direction Method (ADM) [33].

For $n>1$ (time t_n), we use the block coordinate descent method to update $\mathbf{a}^n$ and $\mathbf{B}^{n-1}$ alternately. We reduce (6) to the following two sub-problems. The sub-problem for updating $\mathbf{a}^n$ is to minimize:

$$\begin{aligned}L(\mathbf{a}^n)=&\frac{1}{2}\sum_{m=1}^{M}\int_{\Omega}\left[O_m-u_m^n-\Delta t\ (\mathbf{a}^n)^T\cdot\mathbf{inv}(u_m^n,\mathbf{v}_m^n)\right]^2\mathrm{d}\Omega\\&+\lambda_1\|\mathbf{a}^n\|_1.\end{aligned}\tag{8}$$

It is essentially the same as 7.

When $\mathbf{a}^n$ is fixed, by inserting the second equation of (5) in the first equation in (5), problem (6) reduces to minimizing:

$$\begin{aligned}L(\mathbf{B}^{n-1})=&\frac{1}{2}\sum_{m=1}^{M}\int_{\Omega}\Big[O_m-u_m^n-\Delta t\ (\mathbf{a}^n)^T\cdot\mathbf{inv}(u_m^n,\mathbf{v}^{n-1}\\&+\Delta t\left(\mathbf{B}^{n-1}\right)^T\cdot\mathbf{inv}(u_m^{n-1},\mathbf{v}_m^{n-1})\Big)\Big]^2\,\mathrm{d}\Omega+\lambda_2\|\mathbf{B}^{n-1}\|_1.\end{aligned}\tag{9}$$

This is a highly-nonlinear and non-convex problem. Like [28], we linearize the term in $\int_{\Omega}(\cdot)^2\,\mathrm{d}\Omega$ in (9) locally and then solve it in the same way as (8). $\mathbf{B}^{n-1}$ can be simply initialized as $\mathbf{0}$.

We summarize the whole solution process of learning-based PDEs in Algorithm 1. Here we use Accelerated Proximal Gradient (APG) [32] to solve the sub-problems (8)–(9).

Algorithm 1. Learning-based PDEs.

Input Training image pairs $\{(I_m,O_m)\}_{m=1}^{M}$.
Initialize
$u_m^0=\mathbf{1},\mathbf{v}_m^0=[I_{m,1},I_{m,2},I_{m,3}]^T,\Delta t=0.05,\varepsilon=10^{-6},N=10.$
Step 0 Compute $\mathbf{a}^0$ by solving problem (7) and u_m^1 by using (5).
Step n ($n\geq 1$)
While not converged **do**
$\mathbf{B}^{n-1}=\mathbf{0}$.
While not converged **do**
1. Fix $\mathbf{B}^{n-1}$ and update $\mathbf{a}^n$ by solving sub-problem (8),
2. Fix $\mathbf{a}^n$ and update $\mathbf{B}^{n-1}$ by solving sub-problem (9),
end while
Compute u_m^{n+1} and $\mathbf{v}_m^n$ by using (5),
Check the convergence conditions:
$\|L^{n+1}(\mathbf{a}^n,\mathbf{B}^{n-1})-L^n(\mathbf{a}^{n-1},\mathbf{B}^{n-2})\|_\infty<\varepsilon$ or $n\geq N$,
$n\leftarrow n+1$.
end while

3.1.3. Numerical implementation

We need to prepare the input/output training image pairs (in our experiments, we only use $M=60$ pairs) before learning the coefficients. The input training image is like Fig. 4(a), which is the initial value of $\mathbf{v}$. The initial value of u is taken as 1 at each pixel. The output training image is the text mask image, like Fig. 4(b).

To compute the spatial derivatives and integrations, we need to do spatial discretization. We use central differences to approximate the derivatives:

$$\begin{cases}\dfrac{\partial f}{\partial x}=\dfrac{f(x+1)-f(x-1)}{2},\\[2mm]\dfrac{\partial^2 f}{\partial x^2}=f(x+1)-2f(x)+f(x-1).\end{cases}\tag{10}$$

The discrete forms of $\partial f/\partial y$, $\partial^2 f/\partial y$, and $\partial^2 f/\partial x\partial y$ can be defined similarly. In addition, we discretize the integrations as

$$\int_{\Omega}f(x,y)\,\mathrm{d}\Omega=\frac{1}{|\Omega|}\sum_{(x,y)\in\Omega}f(x,y),\tag{11}$$

where $|\Omega|$ is the number of pixels in Ω.

We choose $\lambda_1=\lambda_2=10^{-3}$ and show the evolution of one image governed by learnt PDEs in Fig. 4. It can be seen that at the beginning steps the text confidence values may not be very high (see Fig. 4(c)). However, the values increase gradually with the evolution of PDEs (see Fig. 4(d)–(f)).

3.2. Text region candidates detection

When given a test image, we compute the confidence map (Fig. 4(f)) by solving the learned PDEs, with the test image being the initial condition. Then we adopt Niblack's local binarization algorithm [15] due to its high efficiency and robustness to image degradation. After binarization, we find out the 8-connected atom components and then group them into text region candidates.

For clustering, we consider the atom components as the vertices V and build an undirected graph $G=(V,E)$. Two vertices are connected if and only if the following three conditions are satisfied. Let (x_i,y_i) denote the top left corner of the bounding rectangle of atom component V_i, h_i and w_i be the height and the width of the bounding rectangle, respectively, and $\{c_{1,i},c_{2,i},c_{3,i}\}$ be the means of three color channel values of V_i. Then the conditions on V_i and V_j are

- The bounding rectangles should align. So
$$\frac{\min\left(|y_i-y_j|,|y_i+h_i-y_j-h_j|\right)}{\min(h_i,h_j)}$$
should be less than a threshold T_1.
- The two atom components should not be too far from each other. So the interval
$$\frac{\min\left(|x_i+x_j-w_i|,|x_i-x_j-w_j|\right)}{\min(w_i,w_j)}$$
should not be greater than a threshold T_2.
- The atom components in the same region need to have similar colors. So the color difference
$$\sqrt{(c_{1,i}-c_{1,j})^2+(c_{2,i}-c_{2,j})^2+(c_{3,i}-c_{3,j})^2}$$
should be lower than a predefined threshold T_3.

In our experiments, we set $T_1=0.5$, $T_2=2$, and $T_3=50$. We find the connected components of the graph G as different clusters. After clustering, we calculate the minimum circumscribed

rectangle as the text region candidates. We further use simple heuristic rules, such as the ratio of the width to the height of the minimum circumscribed rectangle, to remove wrong text region candidates. All thresholds and parameters for the geometric tests were learned on the fully annotated training set ICDAR 2011 [17].

4. Bottom-up: text detection in the region candidates

Although using learning-based PDEs for text confidence map computation is our major contribution, to make our method complete and comparable to others, in this section we propose a simple framework to extract text strings in the detected text region candidates. It can be replaced by any CC-based methods and the performance can be further improved if more advanced CC-based methods are employed. The framework is the bottom-up scheme and is mainly inspired by [21,26]. It consists of three main steps: character candidates extraction, text candidates construction, and final result verification.

4.1. Character candidates extraction

For character candidates extraction, color and edge/gradient features [21] are conventionally used. Recently, stroke width (e.g., SWT [5,25,26]) and region features (e.g., MSER [23,24]) have been widely employed. Here we use two methods, binarization and mean shift [34], which simply use the color features. For binarization, we just adopt the method OSTU [16]. For mean shift, we extract the character candidates based on the color of pixels through the following steps, which is inspired by [21,34,35].

Firstly, we eliminate the edge pixels by applying the Canny detector [36] to our text region candidates and calculate the color histogram on the remaining pixels. We choose 64 initial mean color points and adopt the flat kernel with $\lambda = 16$ for the mean shift procedure [37]. After that a pixel belongs to the cluster of the final generated mean color point only when the distance between them is less than 16. Then the 64 final clusters whose weighted mean color are close enough to each other are merged together, producing the final clusters. In our experiments, we observe that the algorithm converges in about 20 iterations.

After character candidates extraction, we further train a character level classifier using the GML AdaBoost toolbox [38]. It generates a confidence value $p_1(c)$ for each character candidate c. Then a simple thresholding method is used to eliminate the negative ones. More details can be seen in Section 4.3.

4.2. Text candidates construction

There are two general approaches, rule-based methods [39,25,10] and cluster-based methods [12,24], for text candidates construction. The rule-based methods assume that characters in a word can be fit by one or more top and bottom lines and text candidate need to connect three or more character candidates. Pan et al. [12] and Yin et al. [24] cluster the character candidates with a learned distance metric. Here we simply follow the same way as we extract the text region candidates to group the character candidates into text candidates. We consider the character candidates as the vertices V and build an undirected graph $G = (V, E)$. Then we find the connected components of the graph as clusters. To build the edge E between two character candidates (vertices), we check whether the following six conditions are satisfied. The first three are the same as those in Section 3.2. Following the same denotations and denoting $|V_i|$ as the number of points in V_i, we present three more conditions as follows:

- The area of each character in a text line should be similar. So the difference

$$\frac{\mathrm{abs}(|V_i| - |V_j|)}{\min(|V_i|, |V_j|)}$$

should not be greater than T_4.
- Considering the capital and lowercase characters, the height ratio

$$\frac{|h_i - h_j|}{\max(h_i, h_j)}$$

should be less than T_5.
- Considering the different widths of characters, the width ratio

$$\frac{|w_i - w_j|}{\max(w_i, w_j)}$$

should be less than T_6.

In our experiments, we set $T_1 = 0.25$, $T_2 = 3$, $T_3 = 50$, $T_4 = 3$, $T_5 = 0.6$, and $T_5 = 0.6$. After clustering, we calculate the minimum circumscribed rectangle as the text candidates.

4.3. Final result verification

There may be many false positives in text candidates because a small piece of components/patches may not contain sufficient information for classification. Recently, a two-level classification scheme (character candidates classifier and text candidates classifier) is widely used for text verification. Inspired by Yin et al. [24],

Table 2
Comparison with most recent text detection results on the ICDAR 2005 test database. The results of other methods are quoted from [12,26].

Methods	Description	Precision	Recall	F-measure
Our method	**Hybrid**	**0.87**	**0.67**	**0.76**
Pan et al. [12]	Hybrid	0.67	0.70	0.69
Huang et al. [26]	CC (SWT)	0.81	0.74	0.72
Yao et al. [25]	CC (SWT)	0.69	0.66	0.67
Epshtein et al. [5]	CC (SWT)	0.73	0.60	0.66
Chen et al. [46]	CC (MSER)	0.73	0.60	0.66
Neumann and Matas [48]	CC (MSER)	0.65	0.64	0.63
Wang et al. [49]	CC	0.77	0.61	0.68
Yi and Tian [50]	CC	0.71	0.62	0.63
Zhang and Kasturi [51]	CC	0.73	0.62	-
Yi and Tian [21]	CC	0.71	0.62	0.62
Lee et al. [38]	Sliding window	0.66	0.75	0.70

Table 3
Comparison with most recent text detection results on the ICDAR 2011 test database. These results are from the papers [24,47].

Methods	Description	Precision	Recall	F-measure
Our method	**Hybrid**	**0.88**	**0.69**	**0.78**
Yin et al. [24]	CC (MSER)	0.86	0.68	0.76
Neumann and Matas [47]	CC (MSER)	0.85	0.68	0.75
Ye et al. [4]	CC (MSER)	0.89	0.62	0.73
Neumann and Matas [23]	CC (MSER)	0.79	0.66	0.72
Shi et al. [52]	CC (MSER)	0.83	0.63	0.72
Koo et al. [53]	CC (MSER)	0.83	0.63	0.71
Huang et al. [26]	CC (SWT)	0.82	0.75	0.73
Yi and Tian [50]	CC	0.81	0.72	0.71
Wang et al. [49]	CC	0.71	0.57	0.63

Yao et al. [25], and Huang et al. [26], we utilize a two-level classification scheme for the final result verification. The two levels of features, character level features and text level features are as follows:

1. *Character level features*: There are four kinds of features. The first kind is the occupation percentage of the component and the second is the ratio between the height and width. The third is the ratio of the larger value (between width and height) and its stroke distance mean. The last is a 36-dimensional feature generated by the TCD-C descriptors proposed by Huang et al. [26], but without the feature of stroke width values.
2. *Text level features*: There are three different types of descriptors. Firstly, a 55-dimensional feature generated by the geometric TCD-T descriptors proposed by Huang et al. [26] is used. Secondly, a 16-bin HOG [40] is computed from the text candidate of all the components. Inspired by [41], we compute the correlation matrix between the bins, generating a 136-dimensional feature. Thirdly, there are 42 global features, consisting of local energy of Gabor filter [42,19] and statistical texture measure of image histogram [41,19]. In total, they form a 233-dimensional feature.

After choosing the features of character and text candidates we use the GML AdaBoost toolbox to train two independent classifiers, i.e., the character-level classifier and the text-level classifier. They generate a confidence value $p_1(c)$ for each character candidate c and an initial confidence value $p_2(T)$ for each text candidate T. As introduced in Section 4.1, we first use a simple thresholding method to eliminate the negative character candidates. Then we group the positive ones into text candidates. For a text candidate T which contains n character candidates $c_i, i=1,\ldots,n$, the final confidence value is defined as $P(T)=\sum_{i=1}^{n} p_1(c_i)/2n+p_2(T)/2$. The final result verification is produced by simply thresholding the final confidence value.

5. Discussions

In this section, we discuss the connection between our methods and some related work.

5.1. Comparison with existing learning-based PDEs

Recently, Liu et al. [43,13] and Zhao [28] combine the learning strategy with PDE-based methods for image processing. Although both theirs and our work aim at learning a PDEs system, our work is different from them. Liu et al. [43,13] use the L_2-norm regularization for the coefficients, while Zhao et al. [28] utilize the L_∞-norm. Here, we exploit the L_1-norm regularization which is more suitable for color image processing due to the significantly increased number of fundamental differential invariants. At the best we know, we are the first to apply PDEs to text detection.

Fig. 5. Successful examples (top two rows) and failure examples (bottom row) from the ICDAR databases.

Fig. 6. Successful examples (top two rows) and failure examples (bottom row) from the SVT databases.

5.2. Comparison to Pan et al.'s hybrid method

As mentioned before, Pan et al.'s hybrid method [12] proposes a text region detector based HOG pyramid and a boosted cascade classifier to produce a text confidence map. Then they use a local binarization algorithm to get the character candidates from the text confidence map. After that, they use a conditional random fields model to eliminate the non-characters and group text components into text lines/words using an energy minimization method. In comparison, we use learning-based PDEs to compute a text confidence map and detect text region candidates from the map. Then any CC-based methods can work on these region candidates. Experiment on ICDAR 2005 in Section 6 achieves a F-measure which is 0.07 higher than Pan et al.'s [12].

6. Experimental results

In this section, we compare our method with several state-of-the-art methods on a variety of public databases, including the ICDAR 2005 [44,45], ICDAR 2011 [17], and the two street view text databases, i.e., the SVT 2010 database [5] and the SVT 2011 database [6]. The performance of these methods is quantitatively measured by precision (P), recall (R), and F-measure (F). They are computed using the same definitions as those in [44,45] at the image level. The overall performance values are computed as the average values of all images in the database.

The training output images for learning-based PDEs are the text mask binarization maps (see Fig. 4). As only the ICDAR 2011 database gives the mask maps, all the parameters (include the coefficients of PDEs and the classifiers) are learned on the ICDAR 2011 training database. We use the same parameters when testing our method on ICDAR 2005 and the two SVT databases.

6.1. Experiments on the ICDAR databases

ICDAR 2005 [44,45] and ICDAR 2011 [17] have been widely used as the benchmarks for text detection in natural images. The ICDAR 2005 database includes 509 color images with image sizes varying from 307×93 to 1280×960 pixels. It contains 258 images in the training set and 251 images for testing. The ICDAR 2011 database contains 229 training images and 255 testing ones. Both databases are evaluated in the word level, and have 1114 and 1189 words annotated in their test sets, respectively.

The performances of the proposed approach on the two databases are shown in Tables 2 and 3. By comparison with results presented in [26,12,24], our precisions and F-measures are higher while the recalls are comparable with the highest recall values. It demonstrates a significant improvement over representative methods, such as the SWT-based methods [26,25,5] and MSER-based methods [46,24,47]. Since our bottom-up processing is simple and similar to them, our performance improvement could be attributable to learning-based PDEs that produce relatively good text confidence maps.

The top two rows of Fig. 5 show some successful results. They suggest that our system is robust against large variations in text font, color, size, and geometric distortion. The bottom row of Fig. 5 shows some failure cases, including a single large-scale character, highly blurred texts, and unusual fonts.

6.2. Experiments on the SVT databases

We also test our method on the two widely used street view text databases, the SVT 2010 database [5] and the SVT 2011 database [6]. The SVT 2010 database consists of 307 color images of sizes ranging from 1024×1360 to 1024×768 pixels. The SVT 2011 database contains 647 words and 3796 letters in 249 images collected from Google Street View. However, the format of the ground truth of SVT 2011 database [6] is different from those of

Table 4
Comparison with most recent text detection results on the SVT 2010 database. The results of other methods are quoted from respective papers.

Methods	Description	Precision	Recall	F-measure
Our method, trained on the ICDAR 2011 database	**Hybrid**	**0.72**	**0.41**	**0.52**
Yin et al. [24],				
trained on the SVT 2011 database	CC (MSER)	0.66	0.41	0.51
trained on the ICDAR 2011 database		0.62	0.32	0.42
Phan et al. [54]	CC	0.50	0.51	0.51
Epshtein et al. [5]	CC (SWT)	0.54	0.42	0.47

Table 5
Comparison with most recent text detection results on the SVT 2011 database. The results of other methods are quoted from respective papers.

Methods	Description	Precision	Recall	F-measure
Our method, trained on the ICDAR 2011 database	**Hybrid**	**0.65**	**0.39**	**0.49**
Wang et al. [6]	CC	0.67	0.29	0.41
Neumann et al. [39]	CC (MSER)	0.19	0.33	–

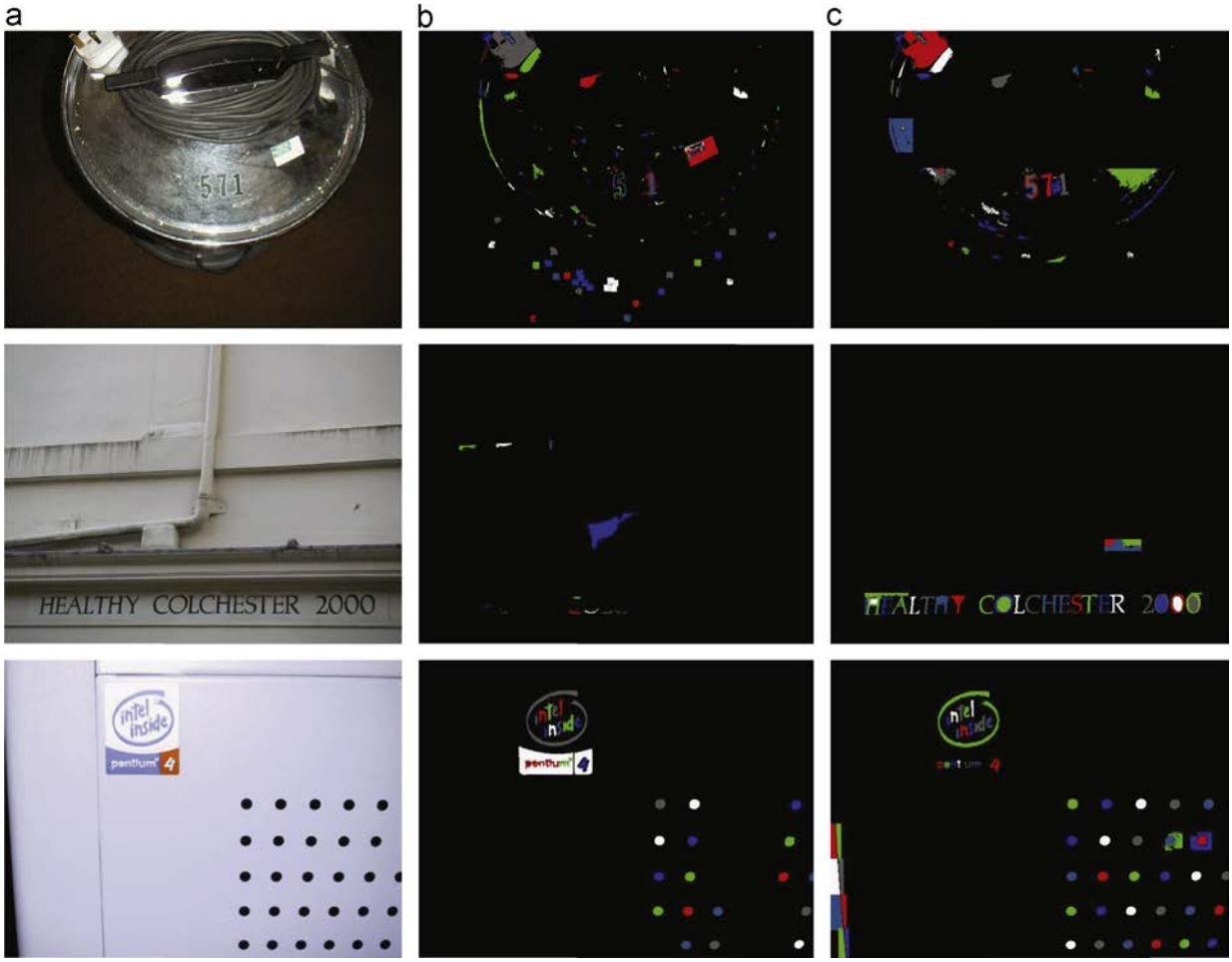

Fig. 7. Comparison of character candidates extraction with and without learning-based PDEs. (a) Original images. (b) Character candidates (in different colors) without learning-based PDEs. (c) Character candidates (in different colors) with learning-based PDEs. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

previous databases as only some words are annotated. Both databases are more challenging than the ICDAR ones. The images in these two database are noisier and they contain many repeating structures such as windows and bricks. Some images are captured in a low light environment and texts tend to be small, blurred, and of low contrast (see Fig. 6). Note that we still use the same

Table 6
Comparison of the results on the ICDAR 2011 database with and without learning-based PDEs.

Scheme	Precision	Recall	F-measure
With learning-based PDEs	**0.88**	**0.69**	**0.78**
Without learning-based PDEs	0.79	0.63	0.70

parameters (including the coefficients of PDEs and the classifiers) trained on the ICDAR 2011 for these two SVT databases.

In Tables 4 and 5, we show the performances on the two SVT databases. It can be seen that our method has a better performance even if it was trained on the ICDAR 2011 database. For the SVT 2011 database, the detection task is very different: localizing words in images (if present), where the words are from a lexicon. Our method has no prior knowledge about the content of the image and its output is not limited by a fixed lexicon. Nonetheless, we still get a higher F-measure of 0.49 using the same evaluation protocol as in the previous section. Fig. 6 shows some detection examples on these challenging databases.

6.3. *Effects of learning-based PDEs*

In this section we show the effectiveness of learning-based PDEs by comparing the results with and without learning-based PDEs. When learning-based PDEs are not utilized to propose text region candidates, we simply apply the bottom-up procedure to the whole image, i.e., we extract the character candidates on the whole image directly (see Fig. 7(b)).

Some comparison examples are shown in Fig. 7. The complexity of the background makes the traditional method difficult to extract the right character candidates on the whole image directly. In contrast, the learning-based PDEs generates region candidates for each image, correctly removing a lot of distractive background. As shown in Fig. 3, some of the text region candidates are so good that we can get correct character candidates simply by image binarization. So we can extract more correct texts, enhancing both the precision and the recall. Table 6 shows the final results comparison on the ICDAR 2011 database with and without learning-based PDEs. All of the precision, recall, and F-measure have been greatly improved if learning-based PDEs are used.

7. Conclusions

In this paper, we propose a novel hybrid approach for text detection in natural scene images. We apply learning-based PDEs to provide text region candidates and devise a simple connected components based method to locate the texts accurately in each text region candidate. Experiment results show the robustness and superiority of our method when compared to many state-of-the-art approaches. In the future, we plan to develop even better operators via learning-based PDEs for robust text region candidates detection from complex backgrounds. We also attempt to apply learning-based PDEs to other high-level vision tasks, such as segmentation and recognition of specific objects and video analysis.

Acknowledgment

Zhenyu Zhao is supported by National Natural Science Foundation of China (NSFC) (Grant no. 61473302). Zhouchen Lin is supported by National Basic Research Program of China (973 Program) (Grant no. 2015CB352502), NSFC (Grant nos. 61272341 and 61231002), and Microsoft Research Asia Collaborative Research Program.

References

[1] Yu Zhong, Hongjiang Zhang, Anil K. Jain, Automatic caption localization in compressed video, IEEE Trans. Pattern Anal. Mach. Intell. 22 (4) (2000) 385–392.
[2] Jerod J. Weinman, Erik Learned-Miller, Allen R. Hanson, Scene text recognition using similarity and a lexicon with sparse belief propagation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (10) (2009) 1733–1746.
[3] Honggang Zhang, Kaili Zhao, Yizhe Song, Jun Guo, Text extraction from natural scene image: a survey, Neurocomputing 122 (2013) 310–323.
[4] Qixiang Ye, David Doermann, Text detection and recognition in imagery: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 37 (7) (2015) 1480–1500. http://dx.doi.org/10.1109/TPAMI.2014.236675.
[5] Boris Epshtein, Eyal Ofek, Yonatan Wexler, Detecting text in natural scenes with stroke width transform, in: CVPR, IEEE, 2010, pp. 2963–2970.
[6] Kai Wang, Boris Babenko, Serge Belongie, End-to-end scene text recognition, in: ICCV, IEEE, 2011, pp. 1457–1464.
[7] Kwang In Kim, Keechul Jung, Jin Hyung Kim, Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 25 (12) (2003) 1631–1639.
[8] Xiangrong Chen, Alan L Yuille, Detecting and reading text in natural scenes, in: CVPR, IEEE, 2004.
[9] Shehzad Muhammad Hanif, Lionel Prevost, Text detection and localization in complex scene images using constrained adaboost algorithm, in: ICDAR, IEEE, 2009, pp. 1–5.
[10] Chucai Yi, YingLi Tian, Text string detection from natural scenes by structure-based partition and grouping, IEEE Trans. Image Process. 20 (9) (2011) 2594–2605.
[11] Céline Mancas-Thillou, Bernard Gosselin, Color text extraction with selective metric-based clustering, Comput. Vis. Image Underst. 107 (1) (2007) 97–107.
[12] Yi-Feng Pan, Xinwen Hou, Cheng-Lin Liu, A hybrid approach to detect and localize texts in natural scene images, IEEE Trans. Image Process. 20 (3) (2011) 800–813.
[13] Risheng Liu, Zhouchen Lin, Wei Zhang, Kewei Tang, Zhixun Su, Toward designing intelligent PDEs for computer vision: an optimal control approach, Image Vis. Comput. 31 (2013) 43–56.
[14] Risheng Liu, Junjie Cao, Zhouchen Lin, Shiguang Shan, Adaptive partial differential equation learning for visual saliency detection, in: CVPR, IEEE, 2014, pp. 3866–3873.
[15] Wayne Niblack, An Introduction to Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ (1986) 115–116.
[16] Nobuyuki Otsu, A threshold selection method from gray-level histograms, Automatica 11 (285–296) (1975) 23–27.
[17] Asif Shahab, Faisal Shafait, Andreas Dengel, ICDAR 2011 robust reading competition challenge 2: reading text in scene images, in: ICDAR, 2011, pp. 1491–1496.
[18] Huiping Li, David Doermann, Omid Kia, Automatic text detection and tracking in digital video, IEEE Trans. Image Process. 9 (1) (2000) 147–156.
[19] Jung-Jin Lee, Pyoung-Hean Lee, Seong-Whan Lee, Alan L Yuille, Christof Koch, Adaboost for text detection in natural scene, in: ICDAR, 2011, pp. 429–434.
[20] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J. Wu, Andrew Y. Ng, Text detection and character recognition in scene images with unsupervised feature learning, in: ICDAR, IEEE, 2011, pp. 440–445.
[21] Chucai Yi, YingLi Tian, Text string detection from natural scenes by structure-based partition and grouping, IEEE Trans. Image Process. 320 (2011) 2594–2605.
[22] Palaiahnakote Shivakumara, Trung Quy Phan, Chew Lim Tan, A Laplacian approach to multi-oriented text detection in video, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2) (2011) 412–419.
[23] Luka Neumann, Jiri Matas, Scene text localization and recognition with oriented stroke detection, in: ICCV, 2013, pp. 97–104.
[24] XuCheng Yin, Xuwang Yin, Kaizhu Huang, HongWei Hao, Robust text detection in natural scene images, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2014) 970–984.
[25] C. Yao, X. Bai, W. Liu, Y. Ma, Z. Tu, Detecting texts of arbitrary orientations in natural images, in: CVPR, 2012.
[26] Weilin Huang, Zhe Lin, Jianchao Yang, Jue Wang, Text localization in natural images using stroke feature transform and text covariance descriptors, in: ICCV, IEEE, 2013, pp. 1241–1248.
[27] E. Zeidler, Nonlinear Functional Analysis and its Applications, Springer-Verlag, Boston, USA, 1985.
[28] Zhenyu Zhao, Zhouchen Lin, Yi Wu, A fast alternating time-splitting approach for learning partial differential equations, Neurocomputing, under review.
[29] Mário A.T. Figueiredo, Robert D. Nowak, Stephen J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, IEEE J. Sel. Top. Signal Process. 1 (4) (2007) 586–597.
[30] Dmitry M. Malioutov, Müjdat Cetin, Alan S. Willsky, Homotopy continuation for sparse signal representation, in: ICASSP, IEEE, vol. 5, 2005, p. v-733.
[31] Patrick L Combettes, Valérie R Wajs, Signal recovery by proximal forward-backward splitting, Multiscale Model. Simul. 4 (4) (2005) 1168–1200.
[32] Amir Beck, Marc Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.
[33] Junfeng Yang, Yin Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing, SIAM J. Sci. Comput. 33 (1) (2011) 250–278.

[34] Dorin Comaniciu, Peter Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.
[35] Nikos Nikolaou, Nikos Papamarkos, Color reduction for complex document images, Int. J. Imaging Syst. Technol. 19 (1) (2009) 14–26.
[36] John Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1986) 679–698.
[37] Keinosuke Fukunaga, Larry Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans. Inf. Theory 21 (1) (1975) 32–40.
[38] Alexander Vezhnevets, GML AdaBoost Matlab Toolbox, Available: ⟨http://graphics.cs.msu.ru/en/science/research/machinelearning/adaboosttoolbox⟩.
[39] Lukas Neumann, Jiri Matas, Real-time scene text localization and recognition, in: CVPR, IEEE, 2012, pp. 3538–3545.
[40] Navneet Dalal, Bill Triggs, Histograms of oriented gradients for human detection, in: CVPR, IEEE, 2005, pp. 886–893.
[41] Jing Zhang, Rangachar Kasturi, Text detection using edge gradient and graph spectrum, in: ICPR, IEEE, 2010, pp. 3979–3982.
[42] Woei Chan, George Coghill, Text analysis using local energy, Pattern Recognit. 34 (12) (2001) 2523–2532.
[43] Risheng Liu, Zhouchen Lin, Wei Zhang, Zhixun Su, Learning PDEs for image restoration via optimal control, in: ECCV, 2010.
[44] Simon M. Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, Robert Young, ICDAR 2003 robust reading competitions, in: ICDAR, IEEE Computer Society, Edinburgh, UK, 2003, pp. 682.
[45] Simon M. Lucas, ICDAR 2005 text locating competition results, in: ICDAR, IEEE, 2005, pp. 80–84.
[46] Huizhong Chen, Sam S. Tsai, Georg Schroth, David M. Chen, Radek Grzeszczuk, Bernd Girod, Robust text detection in natural images with edge-enhanced maximally stable extremal regions, in: ICIP, IEEE, 2011, pp. 2609–2612.
[47] Luka Neumann, Jiri Matas, On combining multiple segmentations in scene text recognition, in: ICDAR, 2013, pp. 523–527.
[48] Lukas Neumann, Jiri Matas, Text localization in real-world images using efficiently pruned exhaustive search, in: ICDAR, IEEE, 2011, pp. 687–691.
[49] Runmin Wang, Nong Sang, Changxin Gao, Text detection approach based on confidence map and context information, Neurocomputing, http://dx.doi.org/10.1016/j.neucom.2015.01.023, in press.
[50] Chucai Yi, Yingli Tian, Text extraction from scene images by character appearance and structure modeling, Comput. Vis. Image Underst. 117 (2) (2013) 182–194.
[51] Jing Zhang, Rangachar Kasturi, Character energy and link energy-based text extraction in scene images, in: ACCV, Springer, Queenstown, New Zealand, 2011, pp. 308–320.
[52] Cunzhao Shi, Chunheng Wang, Baihua Xiao, Yang Zhang, Song Gao, Scene text detection using graph model built upon maximally stable extremal regions, Pattern Recognit. Lett. 34 (2) (2013) 107–116.
[53] Hyung Il Koo, Duck Hoon Kim, Scene text detection via connected component clustering and nontext filtering, IEEE Trans. Image Process. 22 (6) (2013) 2296–2305.
[54] Trung Quy Phan, Palaiahnakote Shivakumara, Chew Lim Tan, Detecting text in the real world, in: Proceedings of the 20th ACM International Conference on Multimedia, ACM, New York, USA, 2012, pp. 765–768.

Zhenyu Zhao received the B.S. degree in mathematics from University of Science and Technology in 2009, and the M.S. degree in system science from National University of Defense and Technology in 2011. He is currently pursuing the Ph.D. degree in applied mathematics, National University of Defense and Technology. His research interests include computer vision, pattern recognition and machine learning.

Cong Fang received the bachelor's degree in electronic Science and technology (for optoelectronic technology) from Tianjin University in 2014. He is currently pursuing the Ph.D. degree with the School of Electronics Engineering and Computer Science, Peking University. His research interests include computer vision, pattern recognition, machine learning and optimization.

Zhouchen Lin received the PhD degree in applied mathematics from Peking University in 2000. Currently, he is a professor at the Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University. He is also a chair professor at Northeast Normal University. He was a guest professor at Shanghai Jiaotong University, Beijing Jiaotong University, and Southeast University. He was also a guest researcher at the Institute of Computing Technology, Chinese Academic of Sciences. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an associate editor of IEEE T. Pattern Analysis and Machine Intelligence and International J. Computer Vision and a senior member of the IEEE.

Yi Wu is a professor in the Department of Mathematics and System Science at the National University of Defense Technology in Changsha, China. He earned bachelor's and master's degrees in applied mathematics at the National University of Defense Technology in 1981 and 1988. He worked as a visiting researcher at New York State University in 1999. His research interests include applied mathematics, statistics, and data processing.