International Journal on Document Analysis and Recognition manuscript No.
(will be inserted by the editor)

Unsupervised refinement of color and stroke features for text binarization

Anand Mishra*, Karteek Alahari, C. V. Jawahar*****

Abstract. Color and strokes are the salient features of text regions in an image. In this work, we use both these features as cues, and introduce a novel energy function to formulate the text binarization problem. The minimum of this energy function corresponds to the optimal binarization. We minimize the energy function with an iterative graph cut based algorithm. Our model is robust to variations in foreground and background as we learn Gaussian mixture models for color and strokes in each iteration of the graph cut. We show results on word images from the challenging ICDAR 2003/2011, born-digital image and street view text datasets, as well as full scene images containing text from ICDAR 2013 datasets, and compare our performance with state-of-the-art methods. Our approach shows significant improvements in performance under a variety of performance measures commonly used to assess text binarization schemes. In addition, our method adapts to diverse document images, like text in videos, handwritten text images.

1 Introduction

Binarization is one of the key preprocessing steps in many document image analysis systems [1, 2]. The performance of subsequent steps like character segmentation and recognition is highly dependent on the success of binarization. Document image binarization has been an active area of research for many years [3–9]. It, however, is not a solved problem in light of the challenges posed by text in video sequences, born-digital (web and email) images, old historic manuscripts and natural scenes where the state-of-the-art recognition performance is still poor. In this context of a variety of imaging systems, designing a powerful text binarization algorithm can be considered a major step towards robust text understanding. Recent interest of the community by organizing binarization contests like DIBCO [10], H-DIBCO [11, 12] at major international document image analysis conferences further highlights its importance.

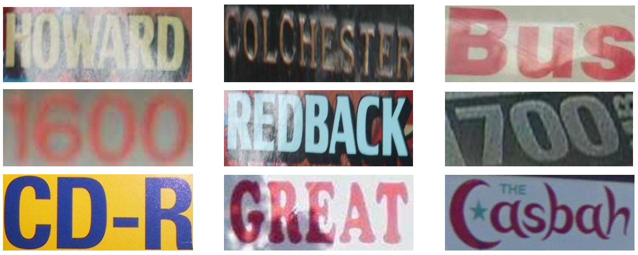

Fig. 1. Sample images we consider in this work. Due to large variations in foreground and background colors, most of the popular binarization techniques in the literature tend to fail on such images (as shown in Section 7).

In this work, we focus on binarization of natural scene text images. These images contain numerous degradations which are not usually present in machine-printed ones, e.g., uneven lighting, blur, complex background, and perspective distortion. A few sample images from the popular datasets we use are shown in Fig. 1. Our proposed method is targeted to such cases, and also to historical handwritten document images.

Our method is inspired by the success of interactive graph cut [13] and GrabCut [14] algorithms for foreground-background segmentation of natural scenes. We formulate the binarization problem in an energy minimization framework, where text is foreground and anything else is background, and define a novel energy (cost)

[*] Anand Mishra
Center for Visual Information Technology
IIIT Hyderabad, India.
E-mail: anand.mishra@research.iiit.ac.in

[**] Karteek Alahari
Thoth team, Inria
Laboratoire Jean Kuntzmann, France.
E-mail: karteek.alahari@inria.fr

[***] C. V. Jawahar
Center for Visual Information Technology
IIIT Hyderabad, India.
E-mail: jawahar@iiit.ac.in

function such that the quality of the binarization is inversely related to the energy value. We minimize this energy function to find the optimal binarization using an iterative graph cut scheme. The graph cut method needs to be initialized with foreground and background seeds. To make the binarization fully automatic, we initialize the seeds by obtaining character-like strokes. At each iteration of graph cut, the seeds and the binarization are refined. This makes it more powerful than a one-shot graph cut algorithm. Moreover, we use two cues to distinguish text regions from background: (i) color, and (ii) stroke width. We model foreground and background colors, as well as stroke widths in a Gaussian mixture Markov random field framework [15], to make the binarization robust to variations in foreground and background.

The contributions of this work are threefold: firstly, we propose a principled framework for the text binarization problem, which is initialized with character-like strokes in an unsupervised manner. The use of color and stroke width features together in an optimization framework for text binarization is an important factor in our work. Secondly, we present a comprehensive evaluation of the proposed binarization method on multiple text datasets. We evaluate the performance using various measures, such as pixel-level and atom-level scores, recognition accuracy, and compare it with the state-of-the-art methods [5,9,16–21] as well as the top-performing methods in the ICDAR robust reading competition [22]. To our knowledge, text binarization methods have not been evaluated in such a rigorous setting in the past, and are restricted to only a few hundred images or one category of document images (e.g., handwritten documents or scene text).

In contrast, we evaluate on more than 2000 images including scene text, video text, born-digital and handwritten text images. Additionally, we also perform qualitative analysis on 6000 images containing video text of several Indian scripts. Interestingly, the performance of existing binarization methods varies widely across the datasets, whereas our results are consistently compelling. In fact, our binarization improves the recognition results of an open source OCR [23] by more than 10% on various public benchmarks. Thirdly, we show the utility of our method in binarizing degraded historical documents. On a benchmark dataset of handwritten images, our method achieves comparable performance to the H-DIBCO 2012 competition winner and a state-of-the-art method [5], which is specifically tuned for handwritten images. The code for our method and the performance measures we use is available on our project website [24].

The remainder of the paper is organized as follows. We discuss related work in Section 2. In Section 3, the binarization task is formulated as a labeling problem, where we define the energy function such that its minimum corresponds to the target binary image. This section also briefly introduces the graph cut method. Section 4 explains the terms of the cost function in detail. In Section 5, we discuss our automatic GMM initialization strategy. Section 6 gives details of the datasets, evaluation protocols, and performance measures used in this work. Experimental settings, results, discussions, and comparisons with various classical as well as modern binarization techniques are provided in Section 7, followed by a summary in Section 8.

2 Related Work

Early methods for text binarization were mostly designed for clean, scanned documents. In the context of images taken from street scenes, video sequences and historical handwritten documents, binarization poses many additional challenges. A few recent approaches aimed to address them for scene text binarization [9,25,26], handwritten text binarization [4,5] and degraded printed text binarization [27]. In this section we review such literature as well as other works related to binarization (specifically text binarization), and argue for the need for better techniques.

We group text binarization approaches into three broad categories: (i) classical binarization, (ii) energy minimization based methods, and (iii) others.

Classical binarization methods. They can be further categorized into: global (e.g., Otsu [17], Kittler [16]) and local (e.g., Sauvola [20], Niblack [19]) approaches. Global approaches compute a binarization threshold based on global statistics of the image such as intra-class variance of text and background regions, whereas local approaches compute the threshold from local statistics of the image such as mean and variance of pixel intensities in patches. The reader is encouraged to refer to [3] for more details of these methods. Although most of these methods perform satisfactorily for many cases, they suffer from problems like: (i) manual tuning of parameters, (ii) high sensitivity to the choice of parameters, and (iii) failure to handle images with uneven lighting, noisy background, similar foreground-background colors.

Energy minimization based methods. Several methods have been proposed for text binarization problems in this paradigm over the last decade [5, 8, 9, 21, 28–32]. Here, the binarization task is posed as an optimization problem, typically modeled using Markov random fields (MRFs). In [21], Wolf and Doermann applied simulated annealing to minimize the resulting cost function. The method proposed in [28], authors first classified a document into text, near text and background regions, and then performed a graph cut to produce the binary image. An MRF based binarization for camera-captured document images was proposed in [29], where a thresholding based technique is used to produce an initial binary image which is refined with a graph cut scheme. The energy function in [29] also uses stroke width as cues, and achieves good performance on printed document images. However, it needs an accurate estimation of stroke width, which is not always trivial in the datasets we use (see Fig. 2). Following a similar pipeline of thresholding followed by labeling with a conditional random field (CRF) model, Zhang *et al.* [30] and Pan *et al.* [31] proposed text extraction methods. These methods however rely on the performance of the thresholding step. Also, being a supervised method, they require large training data with pixel-level annotations for learning a text vs non-text classifier. Hebert *et al.* [32] proposed a scheme where six classical binarization approaches are combined in a CRF framework. Unlike these methods [29–32], our framework does not require thresholding as a first step and proceeds with stroke as well as color initializations which are refined iteratively in an unsupervised manner.

Howe [4] used the Laplacian of image intensity in the energy term for document binarization, and later improved it with a method for automatic parameter selection in [5]. These approaches were designed for handwritten images, and fail to cope up with variations in scene text images, e.g., large changes in stroke width and foreground-background colors within a single image. Adopting a similar framework, Milyaev *et al.* [9] have proposed a scene text binarization technique, where they obtain an initial estimate of binarization with [19], and then use Laplacian of image intensity to compute the unary term of the energy function.

Other methods. Binarization has also been formulated as a text extraction problem [18, 33–36]. Gatos *et al.* [33] presented a method with four steps: denoising with a low-pass Wiener filter, rough estimation of text and background, using the estimates to compute local thresholds, and post-processing to eliminate noise and preserve strokes. Epshtein *et al.* [36] presented a novel operator called the stroke width transform. It computes the stroke width at every pixel of the input image. A set of heuristics were then applied for text extraction. Kasar *et al.* [18] proposed a method which extracts text based on candidate bounding boxes in a Canny edge image. Ezaki *et al.* [34] applied Otsu binarization [17] on different image channels, and then used morphological operators as post processing. Feild and Learned-Miller [37] proposed a bilateral regression based binarization method. This method uses color clustering as a starting point to fit a regression model, and generates multiple hypotheses of text regions. Histogram of gradient features [38] computed for English characters are then used to prune these hypotheses. Tian *et al.* [39] proposed a binarization technique which computes MSER [40] on different color channels to obtain many connected components, and then prune them based on text vs non-text classifier to produce the binarization output. Most of these approaches are either supervised methods requiring large labeled training data, or use multiple heuristics which can not be easily generalized to the diverse datasets we use.

In contrast to the binarization techniques in literature, we propose a method which models color as well as stroke width distributions of foreground (text) and background (non-text) using Gaussian mixture models, and perform inference using an iterative graph cut algorithm to obtain clean binary images. We evaluate publicly available implementations of many existing methods on multiple benchmarks, and compare with them in Section 7.

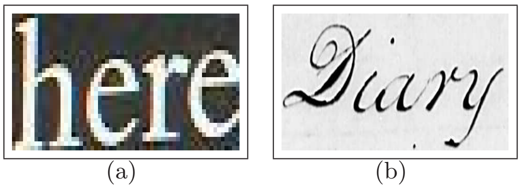

Fig. 2. (a) A scene text image from ICDAR 2013 dataset [22], and (b) part of a handwritten document image taken from H-DIBCO 2012 [11]. We note that stroke width within text is not always constant, and varies smoothly.

This paper is an extension of our initial work [8] which appeared at ICDAR 2011, with the following additions: (i) we initialize candidate text regions using character-like strokes, and refine them in an iterative scheme, instead of relying on a heuristically-designed auto-seeding method, (ii) we incorporate a novel stroke based term in the original color based energy function, and compute its relative importance with respect to color based terms automatically, and (iii) we perform extensive experiments on several recent benchmarks, including handwritten image datasets H-DIBCO 2012/2014, video texts, born-digital images, and ICDAR 2011/2013 datasets.

3 Iterative Graph Cut based Binarization

We formulate the binarization problem in a labeling framework as follows. The binary output of a text image containing n pixels can be expressed as a vector of random variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where each random variable X_i takes a label $x_i \in \{0, 1\}$ based on whether it is text (foreground) or non-text (background). Most of the heuristic-based algorithms take the decision of assigning label 0 or 1 to x_i based on the pixel value at that location, or local statistics computed in a neighborhood. In contrast, we formulate the problem in a more principled framework where we represent image pixels as nodes in a conditional random field (CRF) and associate a unary and pairwise cost for labeling pixels. We then solve the problem by minimizing a linear combination of two energy functions $\boldsymbol{E_c}$ and $\boldsymbol{E_s}$ given by:

$$\boldsymbol{E_{all}}(\mathrm{x}, \boldsymbol{\theta}, \mathrm{z}) = w_1 \boldsymbol{E_c}(\mathrm{x}, \boldsymbol{\theta_c}, \boldsymbol{z_c}) + w_2 \boldsymbol{E_s}(\mathrm{x}, \boldsymbol{\theta_s}, \boldsymbol{z_s}), \qquad (1)$$

such that its minimum corresponds to the target binary image. Here $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ is the set of labels of all the pixels. The model parameters $\boldsymbol{\theta_c}$ and $\boldsymbol{\theta_s}$ are learned from the foreground/background color and stroke width distributions respectively. The vector $\boldsymbol{z_c}$ contains the color values of all the pixels in RGB color space, and the vector $\boldsymbol{z_s}$ contains pixel intensity and stroke width at every pixel.[1] The weights w_1 and w_2 are automatically computed from the text image. To this end, we use

[1] Other color spaces such as CMYK or HSV can also be used.

two image properties, edge density (ρ_1) and stroke width consistency (ρ_2). They are defined as the fraction of edge pixels and standard deviation of stroke widths in the image respectively. We observe that stroke cues are more reliable when we have sufficient edge pixels (i.e., edge density ρ_1 is high), and when the standard deviation of stroke widths is low (i.e., stroke width consistency ρ_2 is low). Based on this, we compute the relative weights ($\hat{w}_1$, $\hat{w}_2$) between color and stroke terms as follows: $\hat{w}_2 = \frac{\rho_1}{\rho_2}$, $\hat{w}_1 = |1 - \hat{w}_2|$. We then normalize these weights to obtain w_1 and w_2 as follows:

$$w_1 = \frac{\hat{w}_1}{\hat{w}_1 + \hat{w}_2}, \tag{2}$$

$$w_2 = \frac{\hat{w}_2}{\hat{w}_1 + \hat{w}_2}, \tag{3}$$

giving more weight to the stroke width based term when the extracted strokes are more reliable, and vice-versa.

For simplicity, we will denote $\boldsymbol{\theta_c}$ and $\boldsymbol{\theta_s}$ as $\boldsymbol{\theta}$ and $\boldsymbol{z_c}$, and $\boldsymbol{z_s}$ as $\mathbf{z}$ from now. It should be noted that the formulation of stroke width based term $\boldsymbol{E_s}$ and color based term $\boldsymbol{E_c}$ are analogous. Hence, we will only show the formulation of color based energy term in the subsequent text. It is expressed as:

$$\boldsymbol{E}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}) = \sum_i E_i(x_i, \boldsymbol{\theta}, z_i) + \sum_{(i,j)\in\mathbf{N}} E_{ij}(x_i, x_j, z_i, z_j), \tag{4}$$

where, $\mathbf{N}$ denotes the neighborhood system defined in the CRF, and E_i and E_{ij} correspond to data and smoothness terms respectively. The data term E_i measures the degree of agreement of the inferred label x_i to the observed image data z_i. The smoothness term measures the cost of assigning labels x_i, x_j to adjacent pixels, essentially imposing spatial smoothness. The unary term is given by:

$$E_i(x_i, \boldsymbol{\theta}, z_i) = -\log p(x_i|z_i), \tag{5}$$

where $p(x_i|z_i)$ is the likelihood of pixel i taking label x_i. The smoothness term is the standard Potts model [13]:

$$E_{ij}(x_i, x_j, z_i, z_j) = \lambda \frac{[x_i \neq x_j]}{dist(i,j)} \quad \exp\left(\beta(z_i - z_j)^2\right), \tag{6}$$

where the scalar parameter λ controls the degree of smoothness, $dist(i,j)$ is the Euclidean distance between neighboring pixels i and j. The smoothness term imposes the cost only for those adjacent pixels which have different labels, i.e., $[x_i \neq x_j]$. The constant β allows discontinuity-preserving smoothing, and is given by: $\beta = 1/2\mathbb{E}[(z_i - z_j)^2]$, where $\mathbb{E}[a]$ is expected value of a [14].

The problem of binarization is now to find the global minima of the energy function $\boldsymbol{E}_{all}$, i.e.,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \boldsymbol{E_{all}}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{z}). \tag{7}$$

The global minima of this energy function can be efficiently computed by graph cut [41] as it satisfies the criteria of submodularity [42]. To this end, a weighted graph $G = (V, E)$ is formed where each vertex corresponds to an image pixel, and edges link adjacent pixels. Two additional vertices source (s) and sink (t) are added to the graph. All the other vertices are connected to them with weighted edges. The weights of all the edges are defined in such a way that every cut of the graph is equivalent to some label assignment to nodes. Here, a cut of the graph G is a partition of the set of vertices V into two disjoint sets S and T, and the cost of the cut is defined as the sum of the weights of edges going from vertices belonging to the set S to T [42, 43]. The minimum cut of such a graph corresponds to the global minima of the energy function, which can be computed efficiently [41].

In [13], $\boldsymbol{\theta}$ corresponds to the parameters of the image foreground and background histograms. These histograms are constructed directly from the foreground and background seeds obtained with user interaction. However, the foreground/background distribution in the challenging cases we target (see images in Fig. 1) cannot be captured effectively by such histograms. Instead, we represent each pixel color (and stroke width) with a Gaussian mixture model (GMM). In this regard, we are inspired by the success of GrabCut [14] for object segmentation. The foreground and background GMMs in GrabCut are initialized by user interaction. We avoid any user interaction by initializing GMMs with character-like strokes obtained using a method described in Section 5.

4 Color and Stroke Width Potentials

The color of each pixel is generated from one of the $2c$ GMMs [44] (c each for foreground and background) with a mean μ and a covariance Σ.[2] In other words, each foreground color pixel is generated from the following distribution:

$$p(z_i|x_i, \boldsymbol{\theta}, k_i) = \mathcal{N}(\mathbf{z}, \boldsymbol{\theta}; \mu(x_i, k_i), \Sigma(x_i, k_i)), \tag{8}$$

where $\mathcal{N}$ denotes a Gaussian distribution, $x_i \in \{0, 1\}$ and $k_i \in \{1, ..., c\}$. To model the foreground color using this distribution, an additional vector $\mathbf{k} = \{k_1, k_2, ..., k_n\}$ is introduced where each k_i takes one of the c GMM components. Similarly, background color is modeled from one of the c GMM components. Further, the overall likelihood can be assumed to be independent of the pixel position, and thus expressed as:

$$p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}, \mathbf{k}) = \prod_i p(z_i|x_i, \boldsymbol{\theta}, k_i), \tag{9}$$

$$= \prod_i \frac{\pi_i}{\sqrt{|\Sigma_i|}} \exp\left(\frac{-\tilde{z}_i{}^T \Sigma_i^{-1} \tilde{z}_i}{2}\right), \tag{10}$$

where $\pi_i = \pi(x_i, k_i)$ is Gaussian mixture weighting coefficient, $\Sigma_i = \Sigma(x_i, k_i)$ and $\tilde{z}_i = (z_i - \mu(x_i, k_i))$. Due to

[2] The stroke-based term is computed similarly with stroke width and intensity of each pixel generated from one of the $2c$ GMMs.

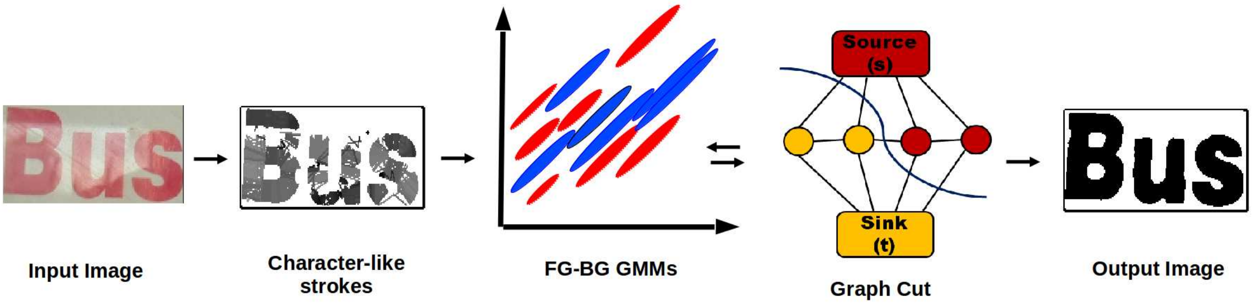

Fig. 3. Overview of the proposed method. Given an input image containing text, we first obtain character-like strokes using the method described in Section 5. GMMs for foreground (text) and background (non-text) are learnt from these initial seeds. We learn two types of GMMs: one using RGB color values and another using stroke width and intensity values. Unary and pairwise costs are computed for every pixel, and are appropriately weighted (see Section 3). An *s*-*t* graph is constructed with these costs. The min cut of this graph produces an initial binary image, which is used to refine the seeds, and the GMMs. The GMM refinement and graph cut steps are repeated a few times to obtain the final binary image. (**Best viewed in pdf**.)

the introduction of GMMs the data term in (4) becomes dependent on its assignment to a GMM component, and is given by:

$$E_i(x_i, k_i, \boldsymbol{\theta}, z_i) = -\log p(z_i|x_i, \boldsymbol{\theta}, k_i). \tag{11}$$

In order to make the energy function robust to low contrast color images we introduce a novel term into the smoothness function which measures the "edginess" of pixels as:

$$E_{ij}(x_i, x_j, z_i, z_j) = \lambda_1 \sum_{(i,j)\in\mathbf{N}} Z_{ij} \; + \lambda_2 \sum_{(i,j)\in\mathbf{N}} G_{ij}, \tag{12}$$

where, $Z_{ij} = [x_i \neq x_j]\exp(-\beta_c||z_i - z_j||^2)$ and $G_{ij} = [x_i \neq x_j]\exp(-\beta_g||g_i - g_j||^2)$. Here g_i denotes the magnitude of gradient (edginess) at pixel i. Two neighboring pixels with similar edginess values are more likely to belong to the same class with this constraint. The constants λ_1 and λ_2 determine the relative strength of the color and edginess difference terms with respect to the unary term, and are fixed to 25 empirically. The parameters β_c and β_g are automatically computed from the image as follows:

$$\beta_c = \frac{1}{\xi} \sum_{(i,j)\in\mathbf{N}} (z_i - z_j)^2, \tag{13}$$

$$\beta_g = \frac{1}{\xi} \sum_{(i,j)\in\mathbf{N}} (g_i - g_j)^2, \tag{14}$$

where $\xi = 2(4wh - 3w - 3h + 2)$ is the total number of edges in the 8-neighborhood system $\mathbf{N}$ with w and h denoting the width and the height of the image respectively.

In summary, both the color and stroke width of foreground and background regions are modeled as GMMs. To initialize these GMMs, we obtain character-like strokes from the given image as described in the following section.

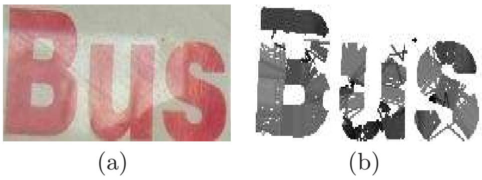
(a) (b)

Fig. 4. (a) Input image. (b) Character-like strokes obtained using the method presented in Section 5. Darker regions in (b) represent parts with lower stroke width.

5 GMM Initialization

Initializing GMMs can play a crucial role as it is hard to recover from a poor random initialization. In this work we propose to obtain initial seeds from character-like strokes. The idea of obtaining character-like strokes is similar in spirit to the work of Epshtein *et al.* [36]. However, unlike [36], our method is robust to incorrect strokes as we refine the initializations iteratively by learning new color and stroke GMMs in each iteration. Alternative techniques can also be used for initialization, such as other binarization techniques [5, 17, 21]. In Section 7.1 we investigate these alternatives empirically.

Obtaining character-like strokes. We begin by extracting an edge image with the Canny edge operator, and then find character-like strokes with the following two-step approach.

We first automatically detect the polarity of the image (see Section 7). If the average gray pixel value in the vertical strip at the center of an image is greater than the average value in the boundary region, we assign a polarity of one (i.e., light text on dark background), otherwise we assign a polarity of zero (i.e., dark text on light background). In the case of images with polarity one, we subtract 180° from the original gradient orientation.

We then detect the strokes in the second step. Let u be an edge pixel with gradient orientation θ. For every such edge pixel u in the image, we trace a line segment along the gradient orientation θ until we find an edge pixel v, whose gradient orientation is $(180° - \theta) \pm 5°$,

Algorithm 1 Overall procedure of the proposed binarization scheme.

```
procedure
Input: Color or grayscale image
Output: Binary image
Initialize:
1. Number of GMM components: $2c$ for color and $2c$ for stroke GMMs.
2. $maxIT$: maximum number of iterations.
3. Seeds and GMMs (Section 5).
4. $iteration \leftarrow 1$
CRF optimization:
  while $iteration \leq maxIT$ do
    5. Learn color and stroke GMMs from seeds (Section 5)
    6. Compute color ($\boldsymbol{E_c}$) and stroke ($\boldsymbol{E_s}$) based terms (Sections 3 & 4)
    7. Construct s-t graph representing the energy (Sections 3 & 4)
    8. Perform s-t mincut
    9. Refine seeds (Section 5)
    10. $iteration \leftarrow iteration + 1$.
  end while
end procedure
```

i.e., the opposite direction approximately. We mark pixels u and v as traversed, and the line segment $\overline{uv}$ as a character-like stroke. We repeat this process for all the non-traversed edge pixels, and mark all the corresponding line segments as character-like strokes.

We use these character-like strokes as initial foreground seeds. Pixels with no strokes are used as background seeds. Fig. 4 shows an example image and the corresponding character-like strokes obtained with the method described. We initialize two types of GMMs: one with color values, and other with stroke width and pixel intensity values, for both foreground and background, from these initial seeds. Note that unlike our previous work [8], (i) we do not use any heuristics to discard some of the strokes, and instead refine this candidate set of strokes over iterations, (ii) background seeds do not need to be explicitly computed, rather, pixels with no strokes are initialized as potential background.

Once the GMMs are initialized, we compute unary and pairwise terms from (11) and (12) for both color and stroke based terms. With the terms in the energy function (1) now defined, iterative graph cut based inference is performed to minimize (1). At each iteration, the initializations are refined, new GMMs are learned from them, and the relative weights between color and stroke terms are recomputed. This makes the algorithm adapt to variations in foreground and background. The overview of our proposed method is illustrated in Fig. 3 and Algorithm 1.

6 Datasets and Performance Measures

To conduct a comprehensive evaluation of the proposed binarization method, we use four scene text, a born-digital text, a video text and two handwritten image datasets. These are summarized in Table 1. In this section, we briefly describe the datasets and their available annotations.

ICDAR cropped word datasets. ICDAR 2003 and ICDAR 2011 robust reading datasets were originally introduced for tasks like text localization, cropped word recognition, and scene character recognition. We use the cropped words from these datasets for evaluating binarization performance. The test sets of these two datasets contain 1110 and 1189 word images respectively [45, 46, 50, 51]. Pixel-level annotations for both these datasets are provided by Kumar *et al.* [52]. Note that pixel-level annotations are available only for 716 images of ICDAR 2011 dataset. We show pixel-level and atom-level results for only these annotated images for this dataset, and refer this subset as ICDAR 2011-S. However, we show recognition results on all the 1189 images of ICDAR 2011.

The ICDAR 2003 dataset also contains a training set of 1157 word images. Pixel-level annotations for these images are provided by [53]. We use 578 word images from this set chosen randomly to validate our choice of parameters (see Section 7.1). We refer to this subset as our validation set for all our experiments.

ICDAR 2013 full scene image dataset. It is composed of outdoor and indoor scene images containing text. There are 233 images in all, with their corresponding ground truth pixel-level text annotations [22].

ICDAR born-digital image dataset (BDI) 2011. Images are often used in emails or websites to embed textual information. These images are known as born-digital text images. As noted in ICDAR 2011 competitions [47], born-digital images: (i) are inherently low-resolution, (ii) often suffer from compression artefacts and severe anti-aliasing. Thus, a method designed for scene text images may not work for these. Considering this, a dataset known as ICDAR born-digital image (BDI) was introduced as part of ICDAR 2011 competitions. It contains 916 word images, and their corresponding pixel-level annotations provided by Kumar *et al.* [52].

Table 1. Datasets used in our experiments.

Dataset	No. of images	Type	Available annotations
ICDAR 2003 word [45]	1110	Scene text	Pixel, text
ICDAR 2011 word [46]	1189	Scene text	Pixel, text
ICDAR 2013 scene text [22]	233	Scene text	Pixel, text
ICDAR BDI 2011 [47]	916	Born-digital	Pixel, text
Street view text [48]	647	Scene text	Pixel, text
CVSI 2015 [49]	6000	Video text	-
H-DIBCO 2012 [11]	14	Handwritten	Pixel
H-DIBCO 2014 [12]	10	Handwritten	Pixel

Street view text. The street view text (SVT) dataset contains images harvested from Google Street View. As noted in [48], most of the images come from business signage and exhibit a high degree of variability in appearance and resolution. We show binarization results on the cropped words of SVT-word, which contains 647 word images, and evaluate it with pixel-level annotations available publicly [52].

Video script identification dataset (CVSI). The CVSI dataset is composed of images from news videos of various Indian languages. It contains 6000 text images from ten scripts, namely English, Hindi, Bengali, Oriya, Gujarati, Punjabi, Kannada, Tamil, Telugu and Arabic, commonly used in India. This dataset was originally introduced for script identification [49], and does not include pixel level annotations. We use it solely for qualitative evaluation of binarization methods.

H-DIBCO 2012/2014. Although our binarization scheme is designed for scene text images, it can also be applied for handwritten images. To demonstrate this we test our method on the H-DIBCO 2012 [11] and 2014 [12] datasets. They contain 14 and 10 degraded handwritten images respectively, with their corresponding ground truth pixel-level annotations.

6.1 Performance Measures

Although binarization is a highly researched problem, the task of evaluating the performance of proposed solutions has received less attention [54]. Due to the lack of well-defined performance measures or ground truth, some of the previous works perform only a qualitative evaluation [55, 56]. This subjective evaluation only provide a partial view of performance. A few others measure binarization accuracy in terms of OCR performance [57]. While improving text recognition performance can be considered as an end goal of binarization, relying on OCR systems which depend on many factors, e.g., character classification, statistical language models, and not just the quality of text binarization, is not ideal. Thus, OCR-level evaluation can only be considered as an indirect performance measure for rating binarization methods [54].

A well-established practice in document image binarization competitions at ICDAR is to evaluate binarization at the pixel level [10]. This evaluation is more precise than the previous two measures, but has a few drawbacks: (i) pixel-level ground truth for large scale datasets is difficult to acquire, (ii) defining pixel accurate ground truth can be subjective due to aliasing and blur, (iii) a small error in ground truth can alter the ranking of binarization performance significantly as studied in [58]. To address these issues, Clavelli *et al.* [54] proposed a measure for text binarization based on an atom-level assessment. An atom is defined as the minimum unit of text segmentation which can be recognized on its own. This performance measure does not require pixel accurate ground truth, and measures various characteristics of binarization methods such as producing broken text, merging characters.

In order to provide a comprehensive analysis, we evaluate binarization methods on these three measures, i.e, pixel-level, atom-level, and recognition (OCR) accuracy.[3]

Pixel-level evaluation. Given a ground truth image annotated at the pixel-level and the result of a binarization method, each pixel in the output image is classified as one of the following: (i) true positive if it is a text pixel in both the output and the ground truth image, (ii) false positive if it is a background pixel in the output image but a text pixel in the ground truth, (iii) false negative if it is a text pixel in the output image but background pixel in the ground truth, or (iv) true negative if it is background pixel in both the output and the ground truth images. With these in hand we compute *precision*, *recall* and *f-score* for every image, and then report mean values of these measures over all the images in the dataset to compare binarization methods.

Atom-level evaluation. Each connected component in the binary output image is classified as one

[3] Source code for all the performance measures used in this work is available on our project website [24].

of the six categories [54], using following two criteria. (i) The connected component and the skeleton[4] of the ground truth have at least θ_{min} pixels in common. (ii) If the connected component comprises pixels that do not overlap with text-area in the ground truth, their number should not exceed θ_{max}. The threshold θ_{min} is chosen as 90% of the total pixels in the skeleton, and the threshold θ_{max} is either half of the maximum thickness of connected components in the image or five, whichever is lower, as suggested by Clavelli *et al.* [54]. Each connected component in the output image is classified into one of the following categories.

- *whole* (w). If the connected component overlaps with one skeleton of the ground truth, and both criteria are satisfied.
- *background* (b). If the connected component does not overlap with any of the skeletons of the ground truth.
- *fraction* (f). If the connected component overlaps with one skeletons of the ground truth, and only criteria (ii) is satisfied.
- *multiple* (m). If the connected component overlaps with many skeletons of the ground truth, and only criteria (i) is satisfied.
- *fraction and multiple* (fm). If the connected component overlaps with many skeletons of the ground truth, and only criteria (ii) is satisfied.
- *mixed* (mi). If the connected component overlaps with many skeletons of the ground truth, and neither criteria (i) nor criteria (ii) is satisfied.

The number of connected components in the above categories is normalized by the number of ground truth connected components for every image to obtain scores (denoted by w, b, f, m, fm, mi). Then the mean values of these scores over the entire dataset can be used to compare binarization methods. Higher values (maximum = 1) for w, whereas lower values (minimum = 0) for all the other categories are desired. Further, to represent atom-level performance with a single measure, we compute:

$$\textit{atom-score} = \frac{1}{\frac{1}{w} + b + f + m + fm + mi}. \tag{15}$$

The *atom-score* is computed for each image, and the mean over all the images in the dataset is reported. The desired mean *atom-score* for a binarization method is 1, denoting an ideal binarization output.

OCR-level evaluation. We use two well-known off-the-shelf OCRs: Tesseract [23] and ABBYY fine Reader 8.0 [60]. Tesseract is an open source OCR whereas ABBYY fine Reader 8.0 is a commercial OCR product. We report word recognition accuracy which is defined as the number of correctly recognized words normalized by the total number of words in the dataset. Following the ICDAR competition protocols [61], we do not perform any edit distance based correction with lexicons, and report case-sensitive word recognition accuracy.

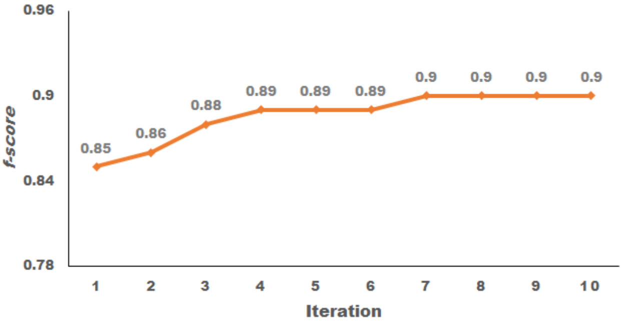

Fig. 5. The pixel-level *f-score* on the subset of ICDAR 2003 training images, used as validation set, at each iteration of graph cut.

7 Experimental Analysis

Given a color or grayscale image containing text, our goal is to binarize it such that the pixels corresponding to text and non-text are assigned labels 0 and 1 respectively. In this section, we perform a comprehensive evaluation of the proposed binarization scheme on the datasets presented in Section 6. We compare our method with classical as well as modern top-performing text binarization approaches with all the performance measures defined in Section 6.1.

7.1 Implementation details

We use publicly available implementations of several binarization techniques for comparison. Global thresholding methods Otsu [17] and Kittler [16] are parameter-independent. For local thresholding methods Niblack [19] and Sauvola [20], we choose the parameters by cross-validating on the ICDAR validation set. For more recent methods like [5,9,18,37] we use the original implementations provided by the authors.[5] For the methods proposed in [5, 9, 37], we use the parameter settings suggested by the corresponding authors. The method in [18] is originally designed for full scene images, and uses heuristics on candidate character bounding boxes. We modify these heuristics, i.e., the maximum allowed height for a character candidate bounding box is changed from 80% of image height to 99% of image height, thereby adapting the method for cropped word images.

Polarity check. Most of the binarization methods in the literature produce white text on black background for images with light text on dark background. Since ground truth typically contains black text on white ground, we perform an automatic polarity

[4] Skeleton, also known as morphological skeleton, is a medial axis representation of a binary image computed with morphological operators [59].

[5] We thank the authors for providing the implementation of their methods.

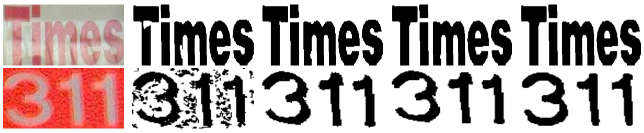

Fig. 6. Illustration of binarization results with different number of iterations of graph cut. Here, we show the original image and the results with 1, 3, 5 and 8 iterations (from left to right).

check before evaluating the method as follows. If the average gray pixel value of the middle part of a given word image is greater than the average gray pixel value of boundary, then we assign reverse polarity, i.e., light text on dark background, to it, and invert the corresponding output image before comparing it with the ground truth. Note that our method produces black text on white background irrespective of the polarity of the word image, and hence does not require this inversion.

It should be noted that handwritten images are always assumed as dark text on light background. Further, we delay the polarity check till the end for full scene images, and obtain the binary images corresponding to both the polarities, i.e., the original image as well as the image where 180° is subtracted from the original gradient orientations. We compute standard deviation of stroke width in both these binary images, and choose the one with lower standard deviation as the final binary image.

We now provide empirical evidence for the choice of parameters, such as, number of iterations, the GMM initialization method, the number of GMMs and weights λ_1 and λ_2 in our method.

Number of iterations. We refine the initial strokes and color cues obtained by our unsupervised automatic initialization scheme. This is performed using iterative graph cuts. To illustrate the refinement of these two cues over iterations, we studied the pixel-level *f-score* on the validation set. This result is shown in Fig. 5. We observe that the pixel-level *f-score* improves with iterations till the seventh, and then remains unchanged. We also show qualitative results over iterations of graph cut in Fig. 6. We note that the iterative refinement using graph cut helps in improving the pixel-level performance. Based on this study, we fix number of iterations to 8 in all our experiments.

GMM initialization. We initialize GMMs by character-like strokes (see Section 5). However, these GMMs can also be initialized using any binarization method. To study its impact, we performed the following experiment. We initialize foreground and background GMMs from three of the best-performing binarization methods in literature: Otsu [17], Wolf [21] and Howe [5], and study the word recognition performance on the validation set. We also studied the effect of user-assisted initialization of foreground and background GMMs. We refer to this as manual initialization (MI). In Fig. 7 we show the word recognition performance of Tesseract on the validation set in two settings: (i) when the above binarization techniques are used, and the binary images are fed to the OCR (lighter gray bars), (ii) when these methods are used for GMM initialization, followed by our iterative graph cut based scheme for binarization, and then the output images are fed to the OCR (darker red bars). We observe that our binarization method improves the word recognition performance irrespective of the initialization used. This is primarily due to the fact that our method iteratively refines the initial seeds by using color and stroke cues, improves the binarization, and subsequently the recognition performance. Further, the variant of our method using manual initialization achieves a high recognition performance on this dataset. This shows that the proposed technique can also prove handy for user-assisted binarization as in [62, 63].

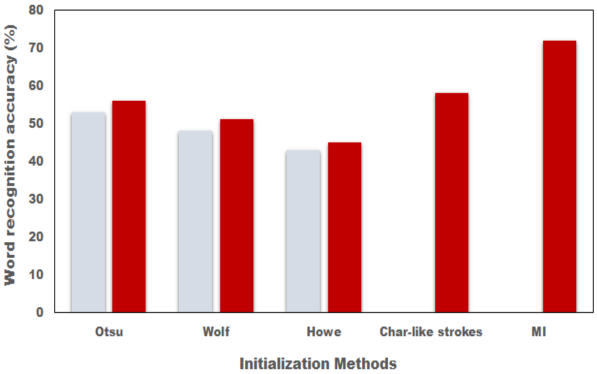

Fig. 7. Impact of GMM initialization techniques. We show the word recognition accuracy of Tesseract on the ICDAR 2003 validation set. Here, lighter (gray) bars show recognition results after applying binarization techniques [5, 17, 21], and darker (red) bars show recognition results of the proposed iterative graph cut based method, with the corresponding binarization techniques used as initialization for GMMs. We also show recognition results when initialization is performed from character-like strokes (char-like strokes) and manually (MI).

Other parameters. We estimate the parameters of our method, i.e., number of color and stroke GMMs (c), and the relative weights between color and edginess terms (λ_1 and λ_2), using grid search on the validation set, and fix them for all our experiments. We vary the number of color and stroke GMMs from 5 to 20 in steps of 5, and compute the validation accuracy (pixel-level *f-score*). We observe only a small change ($\pm$ 0.02) in *f-score* for different numbers of color and stroke GMM. We fix the number of color and stroke GMMs as 5 in all our experiments. We use a similar strategy for choosing λ_1 and λ_2, and vary these two parameters from 5 to 50 in steps of 5. We compute the pixel-level *f-score* on the validation set for all these pairs, and choose the one with the best performance, which results in 25 for both λ_1 and λ_2.

Our method is implemented in C++, and it takes about 0.8s on a cropped word image of size 60 × 180

Table 2. Pixel-level binarization performance. We compare methods on mean *precision*, *recall* and *f-score* values. Here "Ours (color)", "Ours (stroke)" and "Ours (color+stroke)" refer to the proposed iterative graph cut, where only the color, only the stroke, and the color+stroke terms are used respectively. "Ours (MI)" refers to our method with manual initialization of GMMs, and serves as an upper bound.

Method	ICDAR 2003			ICDAR 2011-S			Street View Text		
	precision	*recall*	*f-score*	*precision*	*recall*	*f*-score	*precision*	*recall*	*f-score*
Otsu [17]	0.86	0.90	0.87	0.87	0.91	0.88	0.64	0.83	0.70
Kittler [16]	0.75	0.89	0.78	0.79	0.89	0.80	0.55	0.81	0.62
Niblack [19]	0.68	0.87	0.74	0.75	0.86	0.79	0.52	0.78	0.60
Sauvola [20]	0.65	0.83	0.67	0.73	0.81	0.71	0.52	0.76	0.57
Wolf [21]	0.81	0.91	0.84	0.83	0.90	0.85	0.58	0.81	0.66
Kasar [18]	0.72	0.64	0.65	0.65	0.47	0.52	0.70	0.71	0.69
Milyaev [9]	0.71	0.69	0.63	0.72	0.73	0.65	0.52	0.66	0.51
Howe [5]	0.76	0.84	0.76	0.76	0.87	0.78	0.62	0.77	0.64
Bilateral [37]	0.84	0.85	0.83	0.89	0.87	0.87	0.64	0.79	0.68
Ours (color)	0.82	0.90	0.85	0.86	0.90	0.87	0.62	0.84	0.70
Ours (stroke)	0.78	0.83	0.79	0.80	0.83	0.80	0.63	0.72	0.65
Ours (color+stroke)	0.82	0.91	0.86	0.86	0.91	0.88	0.64	0.82	0.71
Ours (MI)	0.92	0.95	0.93	0.96	0.98	0.97	0.87	0.95	0.90

Table 3. Atom-level evaluation. We show the fractions of connected components classified as *whole*, *background*, *mixed*, *fraction*, and *multiple* categories as well as the *atom-score*. Here "Ours (color)", "Ours (stroke)" and "Ours (color+stroke)" refer to the proposed iterative graph cut, where only the color, only the stroke, and the color+stroke terms are used respectively. "Ours (MI)" refers to our method with manual initialization of GMMs, and serves as an upper bound.

Method	ICDAR 2003						ICDAR 2011-S						Street View Text					
	whole	*background*	*mixed*	*fraction*	*multiple*	*atom-score*	*whole*	*background*	*mixed*	*fraction*	*multiple*	*atom-score*	*whole*	*background*	*mixed*	*fraction*	*multiple*	*atom-score*
Otsu [17]	0.69	2.97	0.06	0.24	0.02	0.59	0.73	1.94	0.04	0.21	0.03	0.63	0.42	0.75	0.08	0.10	0.06	0.34
Niblack [19]	0.50	14.70	0.17	0.74	0.02	0.23	0.57	14.77	0.12	0.85	0.02	0.31	0.35	6.19	0.15	0.20	0.03	0.16
Sauvola [20]	0.37	4.72	0.16	0.44	0.01	0.25	0.44	5.07	0.11	0.63	0.02	0.31	0.26	2.93	0.11	0.33	0.02	0.17
Kittler [16]	0.59	1.34	0.07	0.19	0.04	0.45	0.65	1.05	0.04	0.16	0.04	0.52	0.30	0.59	0.09	0.12	0.05	0.23
Wolf [21]	0.67	3.77	0.08	0.32	0.02	0.56	0.68	1.97	0.06	0.22	0.03	0.58	0.37	1.05	0.12	0.12	0.06	0.28
Kasar [18]	0.51	1.65	0.06	0.34	0.01	0.43	0.38	1.59	0.07	0.33	0.00	0.31	0.49	3.19	0.08	0.26	0.03	0.41
Milyaev [9]	0.36	2.44	0.11	0.37	0.02	0.30	0.37	1.04	0.11	0.30	0.03	0.30	0.27	4.87	0.09	0.18	0.03	0.24
Howe [5]	0.52	0.34	0.11	0.18	0.02	0.46	0.55	0.26	0.10	0.11	0.03	0.50	0.38	13.38	0.09	0.12	0.04	0.32
Bilateral [37]	0.62	2.21	0.08	0.38	0.02	0.52	0.69	2.40	0.04	0.34	0.02	0.60	0.40	5.35	0.09	0.21	0.04	0.31
Ours (color)	0.67	0.58	0.06	0.17	0.03	0.60	0.71	0.38	0.03	0.17	0.03	0.65	0.41	0.75	0.08	0.08	0.07	0.34
Ours (stroke)	0.52	2.75	0.09	0.58	0.01	0.55	0.54	1.95	0.08	0.65	0.01	0.59	0.37	0.69	0.14	0.38	0.02	0.28
Ours (color+stroke)	0.68	0.49	0.06	0.15	0.03	0.62	0.74	0.50	0.04	0.13	0.03	0.67	0.40	0.33	0.08	0.07	0.07	0.34
Ours (MI)	0.77	0.20	0.02	0.13	0.03	0.72	0.86	0.26	0.01	0.09	0.02	0.80	0.64	0.17	0.03	0.09	0.07	0.60

pixels to produce the final result on a system with 2 GB RAM and Intel® Core$^{\mathrm{TM}}$-2 Duo CPU with 2.93 GHz processor system.

7.2 Quantitative Evaluation

Pixel-level evaluation. We show these results in Table 2 as mean *precision*, *recall* and *f-score* on three datasets. Values of these performance measures vary from 0 to 1, and a high value is desired for a good binarization method. We observe that our approach with color only and color+stroke based terms achieves reasonably high *f-score* on all the datasets. The classical method [17] performs better at pixel-level than many other works, and is comparable to ours on the ICDAR 2003 dataset, and poorer on the other two datasets.

Atom-level evaluation. Recall that in this evaluation each connected component in the output image is classified as one of the following categories: *whole*, *background*, *fraction*, *multiple*, *mixed* or *fraction-multiple* (see Section 6.1). Evaluation according to these categories is shown in Table 3. We do not show *fraction-multiple* scores as they are insignificant for all the binarization techniques. Further, we also evaluate binarization methods based on the *atom-score*. An ideal binarization method should achieve 1 for the *atom-score* and the *whole* category, whereas 0 for all other categories. Note that these measures are considered more reliable than pixel-level measures [9, 54].

We observe that our method with color only and color+stroke based terms achieve the best *atom-score*s. On ICDAR 2003 and ICDAR 2011 datasets, our method is ranked first based on the *atom-score*, and improves by 3% and 4% respectively with respect to the next best method [17]. On SVT our method is ranked second. Other recent methods [5, 9, 37] perform well on a few selected images, but fall short in comparison, when tested on multiple datasets.

OCR-level evaluation. OCR results on the ICDAR 2003 and 2011 datasets are summarized in Table 4. We observe that our method improves the performance of OCRs by more than 10% on both these datasets. For example, on the ICDAR 2003 dataset, Tesseract [23] achieves word recognition accuracy of 47.93% without any binarization, whereas when our binarization is applied on these images prior to recognition, the accuracy improves to 56.14%. Our binarization method improves the OCR performance over Otsu by about 5%. Note that all these results are based on case-sensitive evaluation, and we do not perform any edit distance based corrections. It should also be noted that the aim of this work is to obtain clean binary images, and evaluate binarization methods on this performance measure. Hence, we dropped recent word recognition methods which bypass binarization [64–67], in this comparison.

7.3 Qualitative Evaluation

We compare our proposed approach with other binarization methods in Fig. 8. Sample images with uneven lighting, hardly distinguishable foreground/background colors, noisy foreground colors, are shown in this figure. We observe that our approach produces clearly readable binary images with less noise compared to [9, 37]. The global thresholding method [17] performs reasonably well on some examples, but fails unpredictably in cases of high variations in text intensities (e.g., rows 2-3, 7-10). Our method is successful even in such cases and produces clean binary images.

Table 4. Word recognition accuracy (in %): open vocabulary setting. Results shown here are case sensitive, and without minimum edit distance based correction. * No binarization implies that color images are used directly to obtain the corresponding OCR result.

Method	ICDAR 2003		ICDAR 2011	
	Tesseract	*ABBYY*	*Tesseract*	*ABBYY*
No binarization*	47.93	46.51	47.94	46.00
Otsu [17]	51.71	49.10	55.92	53.99
Kittler [16]	44.55	43.25	48.84	48.61
Sauvola [20]	19.73	17.60	26.24	26.32
Niblack [19]	15.59	14.45	22.20	21.27
Kasar [18]	33.78	32.75	12.95	12.11
Wolf [21]	46.52	44.90	50.04	48.78
Milyaev [9]	22.70	21.87	22.07	22.54
Howe [5]	42.88	41.50	43.99	41.04
Bilateral [37]	50.99	47.35	45.16	43.06
Ours (color)	52.25	49.81	59.97	55.00
Ours (stroke)	47.93	46.00	55.75	54.60
Ours (color+stroke)	56.14	52.97	62.57	58.11

7.4 Results on other types of text images

We also evaluate our binarization method on other types of text images, such as, text in videos, born-digital, handwritten text, and full scene images containing text.

For text in videos, we qualitatively evaluate binarization methods on the CVSI dataset [49]. A selection of our results on this dataset are shown in Fig. 9. Despite low-resolution, the performance of our method is encouraging on this dataset. Since our method uses generic text features like color and stroke, which are independent of language, it generalizes to multiple languages as shown in the figure.

We report results on the BDI dataset in Table 5. Our method performs reasonably well, but is inferior to [17] as it suffers from oversmoothing due in part to the extremely low resolution of images in this dataset. This limitation is discussed further in Section 8.

We evaluate on handwritten images of H-DIBCO 2012 [11] and H-DIBCO 2014 [12], and compare the results with other methods for this task. Quantitative results on these datasets are summarized in Table 8. We observe that our proposed method outperforms modern and classical binarization methods, and is comparable to the H-DIBCO 2012 competition winner [5]. On H-DIBCO 2014, our method is marginally inferior to the winning method. Moreover, we achieve noticeable im-

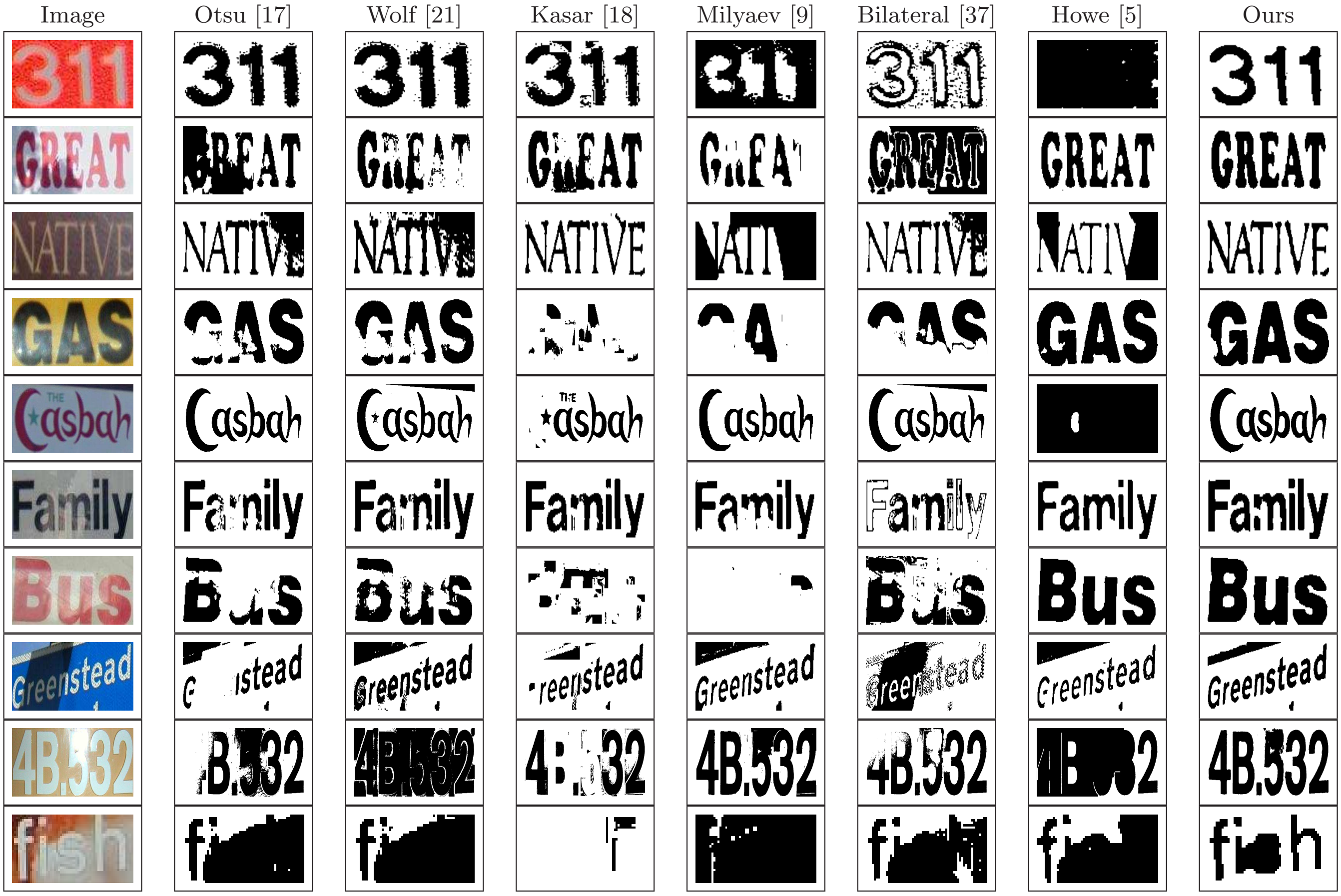

Fig. 8. Comparison of binarization results. From left to right: input image, Otsu [17], Wolf and Doerman [21], Kasar *et al.* [18], Milyaev *et al.* [9], bilateral regression [37], Howe [5] and our method (Ours), which uses color and stroke cues. Other classical techniques [16, 19, 20] show poor performance on these images.

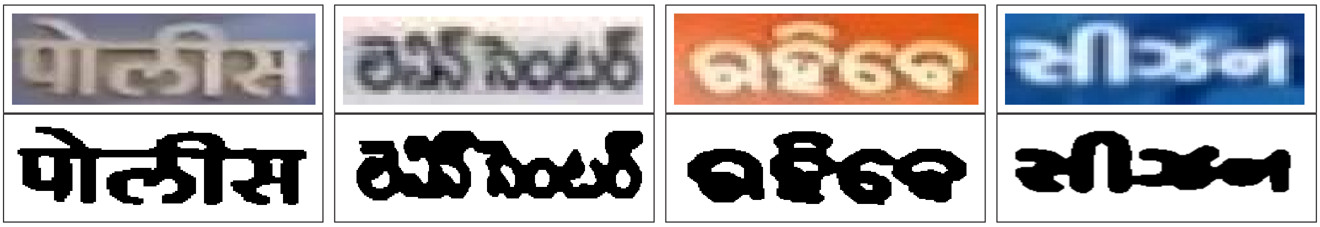

Fig. 9. Results on the CVSI dataset. We show results on images (left to right) with Devanagari, Telugu, Oriya and Gujarati scripts. Since our method does not use any language-specific information, it is applicable to this dataset, containing English, Arabic, and eight Indian scripts.

provement by adding the stroke-based term on these datasets, highlighting their importance for handwritten images. We show qualitative results for a couple of examples in Fig. 10. We observe that despite color bleeding and high variations in pixel intensities and strokes, our method produces a clean binary result. The significance of stroke based term is also highlighted for these examples.

Binarization of natural scene images containing text is a challenging problem. It was considered as one of the challenges in the ICDAR 2013 competitions [22]. Our original work [8] was designed for cropped word images. We now modify our automatic seeding strategy (cf. Section 5) to suit full scene images as well. We evaluate our method on ICDAR 2013, and compare it with the top-performing methods from the competition for the text segmentation task, as shown in Table 6. We compare with the winner method, as well as the first three runner-ups of the competition. Our method with color and stroke terms performs well on this dataset, and stands third in this competition, being marginally inferior to the winner, and comparable to the first runner-up method.

7.5 Comparison with other energy formulations

Energy functions for the binarization task can also be formulated with connected components (CC) or

Table 5. Results on the BDI dataset.

Method	*precision*	*recall*	*f-score*
Otsu [17]	0.77	0.92	0.83
Kittler [16]	0.57	0.88	0.63
Sauvola [20]	0.54	0.94	0.75
Niblack [19]	0.59	0.94	0.71
Kasar [18]	0.55	0.65	0.58
Milyaev [9]	0.48	0.68	0.61
Howe [5]	0.43	0.93	0.52
Bilateral [37]	0.75	0.86	0.79
Ours (color)	0.67	0.88	0.74
Ours (stroke)	0.65	0.80	0.72
Ours (color + stroke)	0.70	0.90	0.80

Table 6. Results on the ICDAR 2013 text segmentation challenge. We compare our method with the top-performing methods in the competition [22].

Method	*precision*	*recall*	*f-score*
The winner (USTB-FuSTAR)	87.21	78.84	82.81
1st runner-up (I2R-NUS)	87.95	73.88	80.31
2nd runner-up (I2R-NUS-FAR)	82.56	73.67	77.86
3rd runner-up (OCTYMIST)	81.82	70.42	75.69
Ours (color)	77.65	71.82	74.89
Ours (stroke)	72.33	68.23	70.45
Ours (color + stroke)	81.00	77.72	79.89

maximally stable extremal regions (MSER) as nodes in the corresponding graph.

Connected component labeling with CRF. We first obtain connected components by thresholding the scene text image using Niblack binarization [19] with the parameter setting in [31]. We then learn an SVM on the ICDAR 2003 training set to classify each component as text or non-text region. Each connected component is represented by its normalized width, normalized height, aspect ratio, shape difference, occupy ratio, compactness, contour gradient and average run-length as in [31]. We then define an energy function composed of a unary term for every CC (computed from the SVM text/non-text classification score), and a pairwise term between two neighboring CCs (truncated sum of squares of the following features: centroid distance, color difference, scale ratio and shape difference). Once the energy function is formulated, we construct a graph representing it, and perform graph cut to label the CCs.

MSER labeling with CRF. We replace the first step of the method described for connected components with MSER, thus defining a graph on MSER nodes, and pose the task as an MSER labeling problem.

Table 7. Comparison with variants, where connected components (CC) and MSER are labeled directly with CRF, instead of labeling our binary segments.

Method	*precision*	*recall*	*f-score*
CC labeling	0.73	0.63	0.65
MSER labeling	0.76	0.82	0.79
Ours (color+stroke)	0.82	0.91	0.86

Table 8. Pixel-level *f-score* on handwritten images from H-DIBCO 2012 and H-DIBCO 2014.

Method	H-DIBCO 2012	H-DIBCO 2014
Otsu [17]	0.75	0.89
Kittler [16]	0.71	0.73
Sauvola [20]	0.14	0.18
Niblack [19]	0.19	0.25
Kasar [18]	0.74	0.78
Wolf [21]	0.78	0.82
Milyaev [9]	0.84	0.92
H-DIBCO 2012 winner [5]	0.89	0.96
H-DIBCO 2014 winner	-	0.97
Ours (Color)	0.84	0.91
Ours (Stroke)	0.78	0.85
Ours (Color+Stroke)	0.90	0.95

Comparison of our method (color+stroke) with these two approaches is shown in Table 7 on the ICDAR 2003 test set. We observe that our method outperforms these two variants whose performance relies extensively on the initialization used. Moreover, extending these two approaches to diverse datasets, such as, handwritten text, text in videos, is not trivial, due to their demand of large pixel-level annotated training sets. On the contrary, our method assigns binary labels to pixels in an unsupervised manner, without the need for such expensive annotation.

8 Summary

In this work we proposed a novel binarization technique, and evaluated it on state-of-the-art datasets. Many existing methods have restricted their focus to small datasets containing only a few images [5, 8, 9, 37]. They show impressive performance on them, but this does not necessarily generalize to the large and varied datasets we consider in this paper. Our method performs consistently well on all the datasets, as we do not make assumptions specific to images. We compare recognition results on public ICDAR benchmarks, where the utility of our work is even more evident. The proposed method integrated with an open source OCR [23] outperforms other binarization techniques (see Table 4). Additionally, on a dataset of video text

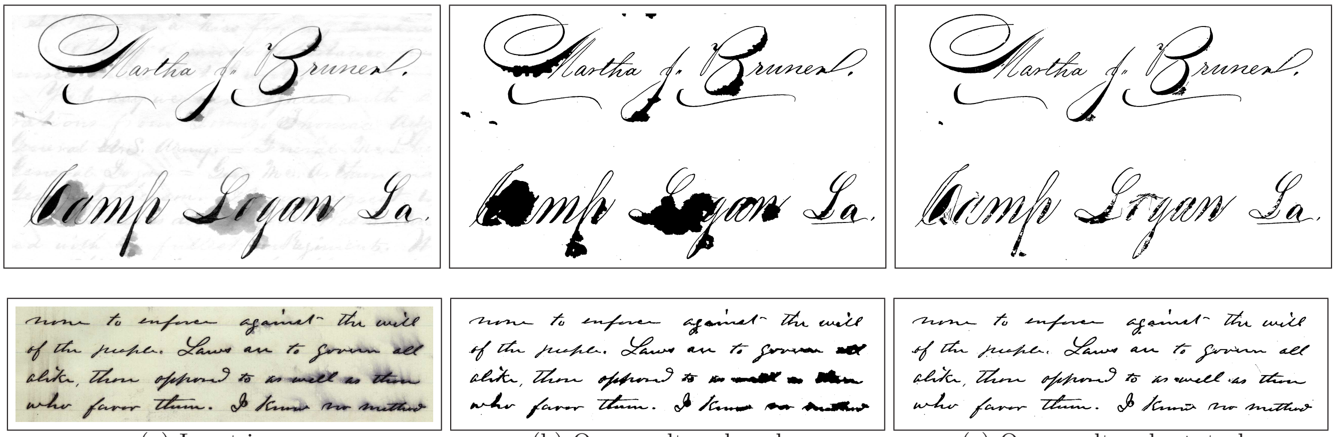
(a) Input image (b) Our result: only color (c) Our result: color+stroke

Fig. 10. Results on sample images from the H-DIBCO 2012 dataset. (a) Input image, and results of our binarization technique: (b) with only color based term, (c) with color and stroke based terms. We observe that the color+stroke based term shows significant improvement over the color only term.

images of multiple scripts, our results are promising, and on two benchmark datasets of handwritten images we achieve results comparable to the state of the art [5, 12].

Comparison with other energy minimization based methods. Some other binarization techniques in the literature are based on an energy minimization framework [5, 9, 21, 28, 29]. Our method falls in this category, but differs significantly in the energy formulation and the minimization technique used. We compare our method empirically with [5, 9, 21] in Tables 2, 3 and 4. Two other energy minimization based methods [28, 29] were dropped for experimental comparison as their implementation was not available when this paper was written. Our method outperforms these approaches. The robustness of our method can be attributed to the proposed iterative graph cut based algorithm, which minimizes an energy function composed of color and stroke based terms.

There have been attempts to solve the natural image segmentation problem using unsupervised iterative graph cut based methods. Jahangiri and Heesch [68] have proposed a method for high contrast natural image segmentation using active contours for initializing the foreground region. In [69, 70] authors use clustering techniques to initialize foreground regions. Our method falls in this category of unsupervised image segmentation, but differs significantly from these approaches in the initialization scheme, and uses text specific information, i.e., character-like strokes, to initialize the foreground regions.

Further improvements. Oversmoothing is one of the limitations of our method, and is pronounced in the case of low resolution images where inter-character gaps and holes within characters like ‘o’, ‘a’ are only a few pixels, i.e., three to four pixels. Such limitations can be handled with techniques like cooperative graph cuts [71]. Further, a noisy automatic initialization may be hard to recover from. Improved initialization or image enhancement techniques can be investigated in future work.

Acknowledgements. This work was partially supported by the Indo-French project no. 5302-1, EVEREST, funded by CEFIPRA. Anand Mishra was supported by Microsoft Corporation and Microsoft Research India under the Microsoft Research India PhD fellowship award.

References

1. Y. Chen and L. Wang, “Broken and degraded document images binarization,” *Neurocomputing (in press)*, 2017.
2. F. Jia, C. Shi, K. He, C. Wang, and B. Xiao, “Document image binarization using structural symmetry of strokes,” in *ICFHR*, 2016, pp. 411–416.
3. P. Stathis, E. Kavallieratou, and N. Papamarkos, “An evaluation technique for binarization algorithms.” *Journal of Universal Computer Science*, vol. 14, no. 18, pp. 3011–3030, 2008.
4. N. R. Howe, “A Laplacian energy for document binarization,” in *ICDAR*, 2011.
5. N. R. Howe, “Document binarization with automatic parameter tuning,” *International Journal on Document Analysis and Recognition*, vol. 16, no. 3, pp. 247–258, 2013.
6. M. Valizadeh and E. Kabir, “Binarization of degraded document image based on feature space partitioning and classification,” *International Journal on Document Analysis and Recognition*, vol. 15, no. 1, pp. 57–69, 2012.
7. G. Lazzara and T. Géraud, “Efficient multiscale Sauvola’s binarization,” *International Journal on Document Analysis and Recognition*, vol. 17, no. 2, pp. 105–123, 2014.
8. A. Mishra, K. Alahari, and C. V. Jawahar, “An MRF model for binarization of natural scene text,” in *ICDAR*, 2011.
9. S. Milyaev, O. Barinova, T. Novikova, P. Kohli, and V. Lempitsky, “Fast and accurate scene text understanding with image binarization and off-the-shelf OCR,” *In-*

ternational Journal on Document Analysis and Recognition, vol. 18, no. 2, pp. 169–182, 2015.
10. I. Pratikakis, B. Gatos, and K. Ntirogiannis, "ICDAR 2013 document image binarization contest (DIBCO 2013)," in *ICDAR*, 2013.
11. I. Pratikakis, B. Gatos, and K. Ntirogiannis, "ICFHR 2012 competition on handwritten document image binarization," in *ICFHR*, 2012.
12. K. Ntirogiannis, B. Gatos, and I. Pratikakis, "ICFHR2014 competition on handwritten document image binarization (H-DIBCO 2014)," in *ICFHR*, 2014.
13. Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in ND images," in *ICCV*, 2001.
14. C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.
15. A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive GMMRF model," in *ECCV*, 2004.
16. J. Kittler, J. Illingworth, and J. Föglein, "Threshold selection based on a simple image statistic," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 2, pp. 125–147, 1985.
17. N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.
18. T. Kasar, J. Kumar, and A. Ramakrishnan, "Font and background color independent text binarization," in *CBDAR*, 2007.
19. W. Niblack, *An introduction to digital image processing.* Strandberg Publishing Company, 1985.
20. J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.
21. C. Wolf and D. Doermann, "Binarization of low quality text using a Markov random field model," in *ICPR*, 2002.
22. D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. Almazán, and L. de las Heras, "ICDAR 2013 robust reading competition," in *ICDAR*, 2013.
23. Tesseract OCR, http://code.google.com/p/tesseract-ocr/.
24. Project website, http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/.
25. C. Thillou and B. Gosselin, "Color binarization for complex camera-based images," in *Electronic Imaging*, 2005.
26. K. Kita and T. Wakahara, "Binarization of color characters in scene images using k-means clustering and support vector machines," in *ICPR*, 2010.
27. S. Lu, B. Su, and C. L. Tan, "Document image binarization using background estimation and stroke edges," *International Journal on Document Analysis and Recognition*, vol. 13, no. 4, pp. 303–314, 2010.
28. J. G. Kuk and N. I. Cho, "Feature based binarization of document images degraded by uneven light condition," in *ICDAR*, 2009.
29. X. Peng, S. Setlur, V. Govindaraju, and R. Sitaram, "Markov random field based binarization for hand-held devices captured document images," in *ICVGIP*, 2010.
30. H. Zhang, C. Liu, C. Yang, X. Ding, and K. Wang, "An improved scene text extraction method using conditional random field and optical character recognition," in *ICDAR*, 2011.
31. Y.-F. Pan, X. Hou, and C.-L. Liu, "Text localization in natural scene images based on conditional random field," in *ICDAR*, 2009.
32. D. Hebert, S. Nicolas, and T. Paquet, "Discrete CRF based combination framework for document image binarization," in *ICDAR*, 2013.
33. B. Gatos, I. Pratikakis, K. Kepene, and S. Perantonis, "Text detection in indoor/outdoor scene images," in *CBDAR*, 2005.
34. N. Ezaki, M. Bulacu, and L. Schomaker, "Text detection from natural scene images: towards a system for visually impaired persons," in *ICPR*, 2004.
35. L. Gomez and D. Karatzas, "A fast hierarchical method for multi-script and arbitrary oriented scene text extraction," *arXiv preprint arXiv:1407.7504*, 2014.
36. B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR*, 2010.
37. J. Feild and E. Learned-Miller, "Scene text recognition with bilateral regression," University of Massachusetts-Amherst, Computer Science Research Center, Tech. Rep. UM-CS-2012-021, 2013.
38. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.
39. S. Tian, S. Lu, B. Su, and C. L. Tan, "Scene text segmentation with multi-level maximally stable extremal regions," in *ICPR*, 2014.
40. J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *BMVC*, 2002.
41. Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.
42. V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.
43. E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 155–225, 2002.
44. D. A. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics, Second Edition*, 2015, pp. 827–832.
45. L. P. Sosa, S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *ICDAR*, 2003.
46. A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," in *ICDAR*, 2011.
47. D. Karatzas, S. R. Mestre, J. Mas, F. Nourbakhsh, and P. P. Roy, "ICDAR 2011 robust reading competition - challenge 1: Reading text in born-digital images (web and email)," in *ICDAR*, 2011.
48. K. Wang and S. Belongie, "Word spotting in the wild," in *ECCV*, 2010.
49. ICDAR 2015 Competition on Video Script Identification. http://www.ict.griffith.edu.au/cvsi2015/.
50. ICDAR 2003 dataset. http://algoval.essex.ac.uk/icdar/RobustWord.html.
51. ICDAR 2011 dataset. http://robustreading.opendfki.de/trac/wiki/SceneText.
52. D. Kumar, M. Prasad, and A. Ramakrishnan, "Benchmarking recognition results on camera captured word image data sets," in *DAR*, 2012.

53. S. Milyaev, O. Barinova, T. Novikova, P. Kohli, and V. Lempitsky, "Image binarization for end-to-end text understanding in natural images," in *ICDAR*, 2013.
54. A. Clavelli, D. Karatzas, and J. Lladós, "A framework for the assessment of text extraction algorithms on complex colour images," in *DAS*, 2010.
55. D. Lopresti and J. Zhou, "Locating and recognizing text in WWW images," *Information Retrieval*, vol. 2, no. 2-3, pp. 177–206, 2000.
56. D. Karatzas and A. Antonacopoulos, "Colour text segmentation in web images based on human perception," *Image and Vision Computing*, vol. 25, no. 5, pp. 564–577, 2007.
57. D. Kumar, M. A. Prasad, and A. Ramakrishnan, "NESP: Nonlinear enhancement and selection of plane for optimal segmentation and recognition of scene word images," in *IS&T/SPIE Electronic Imaging*, 2013.
58. E. H. B. Smith, "An analysis of binarization ground truthing," in *DAS*, 2010.
59. R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Prentice-Hall of India Pvt. Ltd, 2005.
60. ABBYY Finereader 8.0. http://www.abbyy.com/.
61. A. Shahab, F. Shafait, and A. Dengel, "ICDAR2011 robust reading competition challenge 2: Reading text in scene images," in *ICDAR*, 2011.
62. Z. Lu, Z. Wu, and M. S. Brown, "Directed assistance for ink-bleed reduction in old documents," in *CVPR*, 2009.
63. Z. Lu, Z. Wu, and M. S. Brown, "Interactive degraded document binarization: An example (and case) for interactive computer vision," in *WACV*, 2009.
64. A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *CVPR*, 2012.
65. M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *ECCV*, 2014.
66. T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky, "Large-lexicon attribute-consistent text recognition in natural images," in *ECCV*, 2012.
67. C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in *CVPR*, 2013.
68. M. Jahangiri and D. Heesch, "Modified grabcut for unsupervised object segmentation," in *ICIP*, 2009.
69. D. Khattab, H. M. Ebied, A. S. Hussein, and M. F. Tolba, "Multi-label automatic grabcut for image segmentation," in *HIS*, 2014.
70. D. Khattab, H. M. Ebied, A. S. Hussein, and M. F. Tolba, "Color image segmentation based on different color space models using automatic grabcut," *The Scientific World Journal*, vol. 2014, 2014.
71. S. Jegelka and J. Bilmes, "Submodularity beyond submodular energies: coupling edges in graph cuts," in *CVPR*, 2011.