Neurocomputing 304 (2018) 47–63

Contents lists available at ScienceDirect

Neurocomputing

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

A robust proposal generation method for text lines in natural scene images

Kun Fan, Seung Jun Baek*

Department of Computer Science and Engineering, Korea University, Seoul, South Korea

ARTICLE INFO

Article history:
Received 22 May 2017
Revised 20 February 2018
Accepted 22 March 2018
Available online 3 May 2018

Communicated by Dr. Kaizhu Huang

Keywords:
Scene text detection
Feature extraction
Text line proposals
Random Forest

ABSTRACT

Motivated by the success of object proposal generation methods for object detection, we propose a novel method for generating text line proposals from natural scene images. Our strategy is to detect text regions which we define as part of text lines containing a whole character or transitions between two adjacent characters. We observe that, if we scale text regions to a small and fixed size, their image gradients exhibit certain patterns irrespective of text shapes and language types. Based on this observation, we propose simple features which consist of means and standard deviations of image gradients to train a Random Forest so as to detect text regions over multiple image scales and color channels. Text regions are then merged into text line candidates which are ranked based on the Random Forest responses combined with the shapes of the candidates, e.g., horizontally elongated candidates are given higher scores, because they are more likely to contain texts. Even though our method is trained on English, our experiments demonstrate that it achieves high recall with a few thousand good quality proposals on four standard benchmarks, including multi-language datasets. Following the One-to-One and Many-to-One detection criteria, our method achieves 91.6%, 87.4%, 92.1% and 97.9% recall on the ICDAR 2013 Robust Reading Dataset, Street View Text Dataset, Pan's multilingual Dataset and Sampled KAIST Scene Text Dataset respectively, with an average of less than 1250 proposals.

1. Introduction

Texts within an image are valuable because they provide useful semantic information of that image. Such information can be further processed to assist computers and humans to understand, annotate and retrieve images. Recently, textual information understanding have been successfully applied to a wide range of applications, such as vehicle license plate detection and recognition, house numbers detection and recognition, traffic signs detection and translation, and text based image retrieval [1–6]. The process of text understanding from natural scene images is typically divided into two steps: text detection followed by text recognition. We focus on text detection which however is a challenging problem in computer vision. Difficulties of text detection arise due to appearance variations of text and complex backgrounds in real-world environments. When processed against cluttered and complex backgrounds, text variability in terms of different fonts, colors, scales and orientations has deleterious effects on the detection accuracy. For example, texts having rare fonts, different aspect ratios, orientations, non-uniform lighting conditions, partial occlusion or shadowing make detection a challenging task. Background objects such as signs, windows and bricks are similar to true text, and thus can cause false positive detections. Even the state-of-the-art approaches were reported to obtain recall less than 80% [7]. Examples of texts in natural scene images are shown in Fig. 1. In this paper, we consider a *text line proposal generation* method. Our goal is to produce a small number of good quality proposals which cover most of the texts in an image. The extracted proposals can be further processed in the subsequent recognition stage, which is beyond the scope of this paper.

Recently, object proposal generation methods have been widely used in a number of computer vision tasks, e.g., Selective Search [8], BING [9] and Edge Boxes [10]. These methods provide a small set of category-independent candidate bounding boxes which contain most of the objects in an image. The number of bounding boxes typically ranges from hundreds to a few thousands. Thus, the number of bounding boxes is considerably smaller than the total number of sliding windows. As a result, not only the overall processing time is reduced, but also the accuracy is improved by allowing more advanced machine learning algorithms in the subsequent stages. The proposal generation methods have been successfully applied to object detection, and have been reported to

* Corresponding author.
E-mail addresses: kunmoonday@yahoo.com (K. Fan), sjbaek@korea.ac.kr (S.J. Baek).

https://doi.org/10.1016/j.neucom.2018.03.041
0925-2312/

Fig. 1. Some sampled natural scene text images.

perform well in ImageNet detection challenge [11] and PASCAL VOC dataset [12].

Jaderberg et al. [13] applied the object proposal generation methods to natural scene text detection. The authors used a combination of the Edge Boxes algorithm and aggregate channel features detector [14] to generate a set of text proposals. The combination reduced the search space to about 10,000 bounding boxes, which greatly facilitated the subsequent detection and recognition processes. However, a direct application of Edge Boxes to text detection can be problematic. Edge Boxes assumes that objects have a well-defined closed contour, and uses the number of contours which are wholly enclosed in a window so as to derive the probability that the window contains an object. However, text lines or words consist of a sequence of separate letters or strokes; thus, they do not have well-defined boundaries. As a result, different objects, separate letters or strokes are more likely to be extracted than texts. In [13], more than 10,000 proposals were extracted per image. We would like to investigate how to reduce the number of proposals to several hundreds or a few thousands in a holistic way while maintaining comparable recall.

In this paper, we propose a novel text line proposal generation method. Similar to object proposal generation, our goal is to generate a moderate number of high quality language-independent text line proposals with high recall and computational efficiency. We provide a brief overview of our approach as follows. Our method utilizes a small and fixed size window to find text regions. For each window, we estimate the probability that the window contains text. Next, horizontally aligned text regions are grouped into text line candidates, and each text line candidate is given a text score to measure how likely it contains text. Non-maximum suppression (NMS) is used to merge highly overlapped text line candidates and suppress duplicate detections. Finally, text line candidates with top scores are selected as proposals. The above mentioned procedure is repeated at multiple image scales to detect text of different scales.

The novelty of our approach is explained as follows. We focus on generating text line proposals with a good generalization ability to unseen language types. We observe that, if we scale image windows containing text regions of shapes and languages to a small and fixed size, their image gradients exhibit certain patterns. We design simple features to capture these patterns to effectively distinguish text regions from backgrounds. Next, we design a new proposal generation and ranking strategy; we use a fixed size sliding window to search for text regions, and group horizontally aligned regions into text line candidates. Each candidate is scored based on the responses and shapes of aligned regions, e.g., horizontally elongated candidates are given higher scores, because such candidates are more likely to contain text lines. As a result, our method is able to detect text lines irrespective of specific language types, as opposed to traditional sliding window methods which are limited to detecting texts of a single language which they have been trained on.

Through experiments we verify that our method obtains high recall with a few thousand high quality proposals. We validate the performance of the proposed method on a number of datasets, e.g., ICDAR 2013 robust reading dataset [15], Street View Text (SVT) dataset [16], Pan's Multilingual dataset [17] and sampled images from KAIST Scene Text dataset [18]. Our method achieves 91.6% recall on the ICDAR 2013 dataset and 87.4% recall on the SVT dataset with an average of less than 1,250 high quality proposals per image. Moreover, our method is shown to have good generalization ability on two unseen language datasets, e.g., Chinese and Korean datasets (our method is trained on English dataset), which results in high performance under the same experimental setting: 92.1% recall on Pan's multilingual dataset and 97.9% recall on randomly sampled images from the KAIST scene text dataset with a few thousand proposals. The experimental results demonstrate that our method is robust to noise and has good generalization ability to unseen language types. We compare our method with the state-of-the-art generic object proposal methods such as Edge Boxes and Selective Search through experiments, and the results show that our method outperforms those schemes.

The remainder of this paper is organized as follows. In Section 2, we briefly review related work. In Section 3, we describe the proposed method in detail. In Section 4, we present experimental results. We conclude our work in Section 5.

2. Related work

In this section, we will review some of the previous works related to our method.

Text detection: A number of text detection methods have been proposed in recent years; for an overview of existing methods, readers are referred to survey papers, e.g., [7,19,20].

The detection of natural scene texts can be categorized into Connected Component (CC) based methods [21–24] and sliding window based methods [25–29]. The CC based methods consist mainly of three steps: (1) Extraction of CCs which segments pixels (e.g., using pixel intensity values) into CCs, (2) Classifying CCs into characters or background, and (3) Grouping of text CCs into words or text lines. Amongst various approaches, Extremal Regions (ER) [23], Stroke Width Transform (SWT) [21], and Maximally Stable Extremal Regions (MSER) [30] are frequently used. The CC based

Fig. 2. Our method achieves 100% recall with 180 proposals in these testing images, where true positive proposals are marked as red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

methods are widely adopted due to their efficiency, stability and insensitivity to variations in scale, orientation and font. However, the following limitations exist: CC based methods assume that each of the characters can be extracted separately from images. In general, this assumption is not true. For instance, texts in the LEDs as shown in Fig. 2, characters with multiple disconnected strokes (Chinese and Korean characters) and partially occluded characters as shown in Fig. 1 fail to be extracted as a whole but as fragments of a character. Second, low level features used to extract CCs are sensitive to noise, blurring and non-uniform lighting. Third, CC based methods generate a large number of non-text CCs and repeated CCs. Thus, the problem of how to efficiently filter out non-text CCs and to reduce the number of repeated CCs should be addressed.

Sliding window based methods represent another popular class of text detection techniques. Compared to CC based methods, sliding window methods are less sensitive to noise, blurring and non-uniform lighting, but is more computationally demanding. Adaboost [31], Support Vector Machine (SVM) [32] combined with features such as Histogram of Oriented Gradient [33], Local Binary Pattern [34] are widely use in text detection. Recently, a number of text detection methods based on deep Convolutional Neural Network (CNN) [35,36] have been proposed. A CNN [29,37,38] is first trained to classify windows containing instances of characters (26 letters and 10 digits) and backgrounds, then the trained CNN scores every window in a test image. Local maxima are grouped into words or text lines. The CNN-based methods have shown superior performances. However, because the number of window can easily reach an order of 10^6, the main drawback of sliding window methods is the high computational cost. Moreover, it is hard to generalize these methods to unseen language types.

Object proposal methods: The goal of object proposal methods is to generate a moderate number of object-independent candidate bounding boxes which cover many objects in the image. For a detailed review, readers are referred to [39].

Object proposal methods can be divided into two categories: segmentation type and windows scoring type. Segmentation based methods generate segments which are likely to be objects, e.g., Selective Search [8] combines a variety of hand-crafted features to merge pixels into superpixels. The method has been used by state-of-the-art detectors such as R-CNN [40] and Fast R-CNN [41]. Windows scoring based methods use sliding windows, and assign high scores to the windows which are likely to contain an object. Objectness [42,43] finds proposals from salient locations in an image, and scores proposals based on multiple features such as color, edges, locations, etc. BING uses linear SVM classifiers over simple gradient features to score windows. Edge Boxes locates object bounding box proposals from object boundary, and measures the objectness scores based on the number of contours which are wholly enclosed in a window.

Although object proposal methods are originally used to perform object detection, Jaderberg et al. [13] have leveraged the object proposals method in scene text detection. They use Edge Boxes and the aggregate channel feature detector to generate proposals, and refine the proposal quality by using CNN regression. Unlike BING and Edge Boxes which directly score each sliding window, we first find regions corresponding to text lines, then group and score those regions based on a number of new cues. Our method achieves comparable recall but with less number of proposals, and outperforms Edge Boxes and Selective Search on various datasets.

3. The proposed method

An overview of our text line proposal generation method is shown in Fig. 3. Our proposal generation method for query image I consists of the following two stages:

- Top-Down: We use a 12×12 sliding window to scan over three color channels of I, namely *Gray*, C_B and C_R (C_B and C_R are from the YC_BC_R color space), to obtain three response maps $\{R^{Gray}, R^{C_B}, R^{C_R}\}$, respectively. Specifically, feature vectors are extracted from each image window, then a Random Forest is applied to estimate the probability of the window being text region.
- Bottom-Up: We generate and score text line candidates from each response map. Given a response map, horizontally aligned windows which are likely to be text regions are grouped into text line candidates, and are assigned with a text score. Then a maximum of T top scored text line candidates are chosen to build a proposal set. Thus, T represents the maximum number of bounding boxes that can be extracted from each scale. This process is repeated over multiple image scales in order to detect text of different sizes. NMS is applied to merge and suppress duplicate detections.

3.1. Feature extraction

A challenging aspect of proposal generation for text lines in scene images is the variability of text such as font, size, orientation, deformation and language types, etc., as shown in Fig. 4. It is important to find a characterization of text regions which are insensitive to the aforementioned variability. Texts and corresponding backgrounds tend to have different colors, which results in strong edges at the boundary of texts and corresponding backgrounds. As shown in Fig. 5, the English and the Chinese characters are very different from each other in terms of color, shape, language types etc. However, when texts are scaled to a small size, their image gradients have the following patterns. After resizing the text images into a small size (e.g., 12×12) as shown in Fig. 6, we made the following empirical observations on the intensity statistics:

(a) the horizontal derivatives and corresponding standard deviation (STD) are small in the top and bottom regions of text images – backgrounds tend to have a different uniform color;

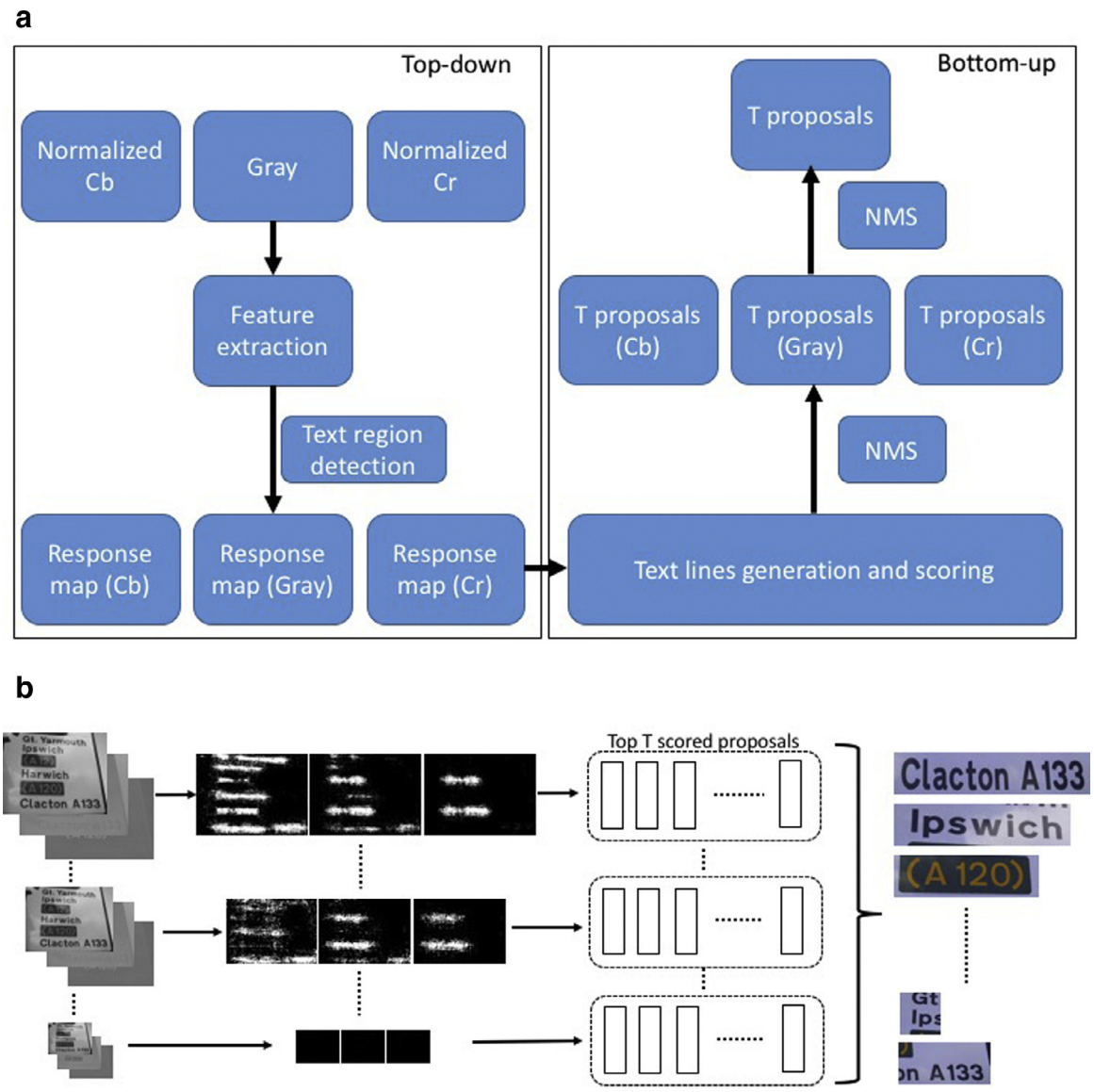

Fig. 3. Overview of the proposed method. (a) Diagram of the proposed text line proposal method. (b) Example of the proposed method. We use a small and fixed size window of dimension 12×12, to scan over images of different scales and channels to get three response maps. Next, we aggregate axis pixels to form text line level candidates, and assign a score to each of the text line candidates. A maximum of top T scored text line candidates are selected as proposals from each scale.

(b) the horizontal derivatives and corresponding STD of are large in the left and right regions of text images, due to the presence of vertical edges and position variations of texts;
(c) the horizontal derivatives have an approximately concave shape, which means the horizontal derivations are large in central area and small in the top and bottom regions;
(d) the STD of horizontal derivatives have an approximately uniform distribution in the central regions, perhaps because after resizing the text images into a small size, the text regions become blurred and have similar intensity values;
(e) the vertical derivatives and corresponding STD are large in the top and bottom regions of text images, owing to the presence of horizontal edges and position variations of texts;
(f) the vertical derivatives and corresponding STD are small in the central region, because text regions become blurred, which reduces intensity variations.

Based on the aforementioned observation, we propose the following features to effectively capture such patterns. Given an image $I(x, y) \in \mathcal{R}^{12\times12}$, we denote the magnitude of image gradients in the horizontal and vertical directions by $g_h(x, y)$ and $g_v(x, y)$, respectively. We divide g_h into 6 equal sized horizontal rectangles (2×12) and 6 equal sized vertical rectangles (12×2), and number the rectangles from 1 to 12, as shown in Fig. 7. For rectangle $j \in \{1, 2, 3, \ldots, 12\}$, we compute μ_h^j which is the arithmetic average of $g_h(x, y)$ over rectangle j as follows:

$$\mu_h^j = \frac{1}{N} \times \sum_{(x,y)\in \text{rectangle}_j} g_h(x, y), \tag{1}$$

where N (in our case $N = 24$) is the number of pixels in rectangle j. The STD θ_h^j of $g_h(x, y)$ over rectangle j is computed as follow:

$$\theta_h^j = \sqrt{\frac{1}{N} \times \sum_{(x,y)\in \text{rectangle}_j} g_h^2(x, y) - (\mu_h^j)^2}. \tag{2}$$

This process is repeated for $j = \{1, 2, 3, \ldots, 12\}$ to obtain a 24 dimensional horizontal gradient feature vector $v_h = \{\mu_h^1, \theta_h^1, \ldots, \mu_h^{12}, \theta_h^{12}\}$.

Following the same schedule, we compute another 24 dimensional vertical feature vector $v_v = \{\mu_v^1, \theta_v^1, \ldots, \mu_v^{12}, \theta_v^{12}\}$ from the

Fig. 4. Natural scene texts exhibit high variations in terms of fonts, texture, size, language types.

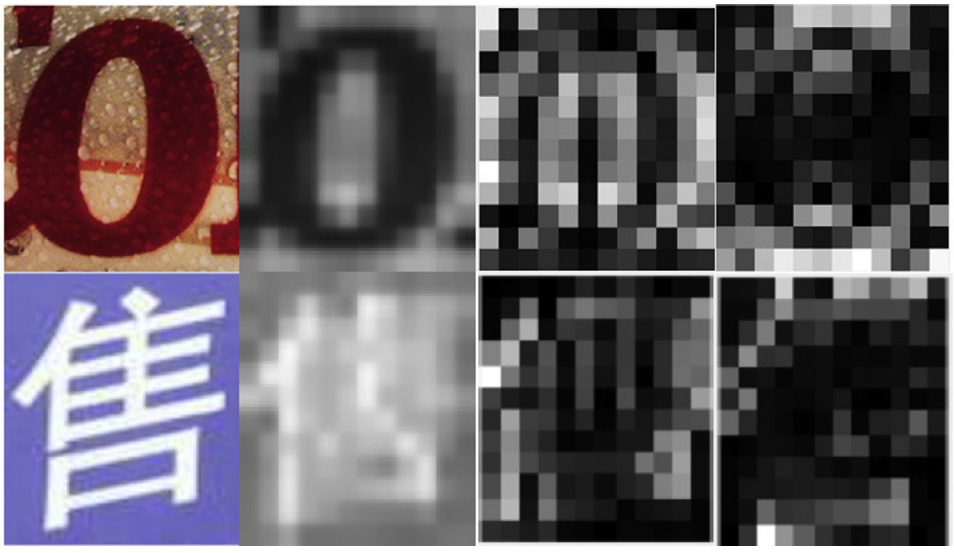

Fig. 5. From left columns to right columns, original images, rescaled images (12×12), horizontal and vertical image gradients of the rescaled text images. Although these two text images come from different languages, after rescaling them to a small size, the image gradients exhibit similar patterns.

$g_v(x, y)$, where μ_v^i and θ_v^i denote the mean and STD of vertical gradient over rectangle j, respectively. Next we concatenate v_h and v_v to obtain a 48 dimensional feature vector to describe the image.

The proposed feature descriptor has several advantages. Firstly, it consists of means and STDs of image gradients, which can be efficiently computed by convolving mean filters with the image gradients, as shown in Algorithms 1 and 2. Specifically, the features can be computed by convolving the image gradients with 12 predefined convolutional filters. The weights of the convolutional filters are set to either 0 or $\frac{1}{24}$. For instance, the weights of the ith block of the ith convolutional filter are set to $\frac{1}{24}$, where the rest of weights are set to 0. The means can be computed by convolving the pre-defined convolutional filters with the image gradients. The STD can be computed by convolving the pre-defined convolutional filters with the squared image gradients, then subtracting the squared means, then taking the square root. The computational complexity of the feature extraction is $O(n)$, where n represents the number of windows in an image.

Secondly, we find that scaling texts to a small size can reduce the interclass variability among texts. Thus, the proposed feature descriptor has good generalization ability to unseen language types, and is robust to deformation of texts. We have applied the proposed features to detect text of different languages, such as Chinese and Korean. Even though Chinese and Korean characters differ

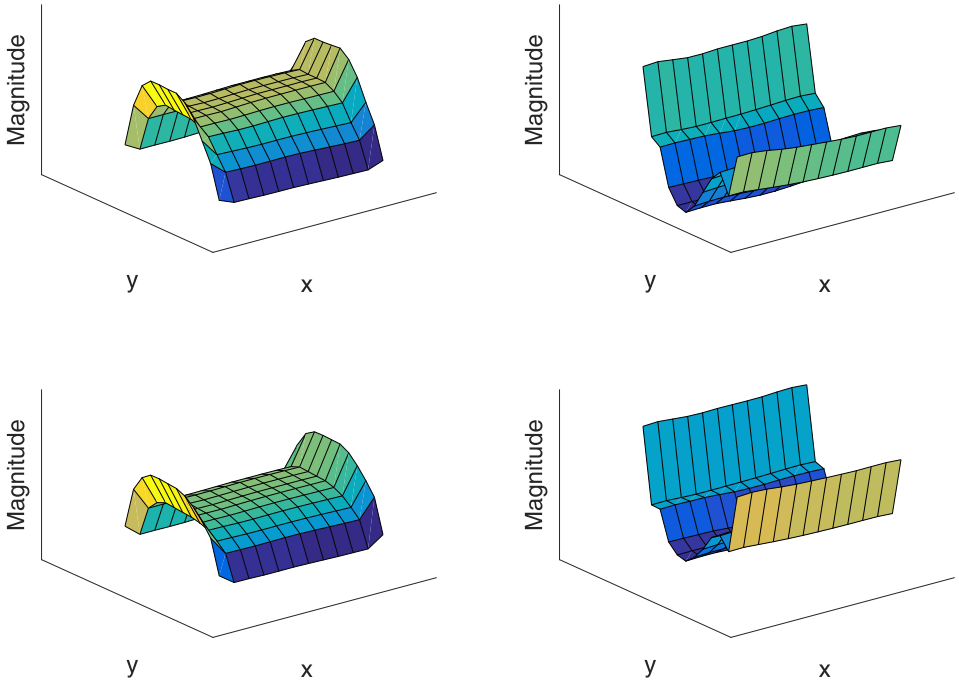

Fig. 6. Gradient analysis of the positive training images. Top-Left: means of horizontal gradient. Top-Right: means of vertical gradient. Bottom-Left: STD of horizontal gradient. Bottom-Right: STD of vertical gradient.

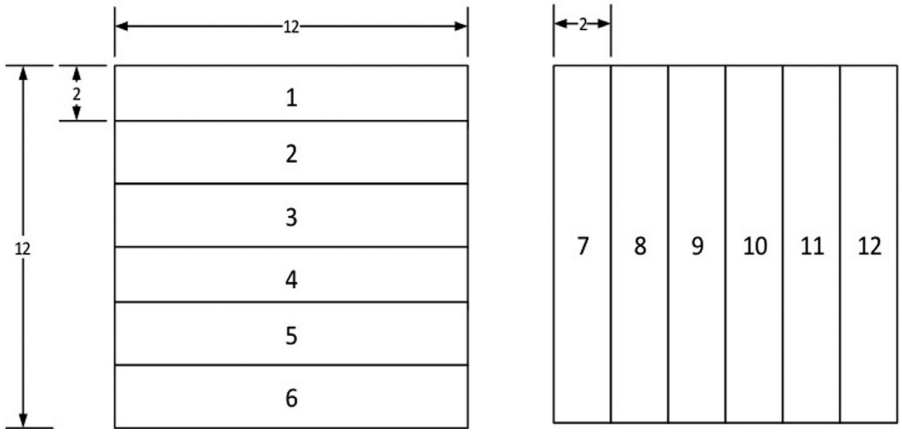

Fig. 7. Rectangular template used for features extraction. The mean features can be computed by convolving 12 mean filters with image gradients and the STD features can be computed by convolving 12 mean filters with the squared image gradients as shown in Eqs. (1) and (2).

significantly from English characters, our method is able to obtain high recall without retraining our classifier with Chinese or Korean datasets, as we discuss in Section 4.

The proposed features are inspired by previous work [25] which uses gray level means, variances and first order differential features as a subset of features for English text detection. However, there are major differences between our method and that in [25]. Firstly, the goal of [25] is English texts detection in scene images which aims at achieving high recall and high precision. However, our method focuses on generating text line proposals achieving good generalization to unseen languages, high computational efficiency and high recall with a manageable number of proposals. That is, our main goal is to reduce the size of search space without performance degradation. Secondly, we find that the means and the STDs of vertical gradient of positive training images are large in the top and bottom regions and low in the central region as shown in Fig 6. This pattern is different from [25] which find the variance of horizontal image gradient is small everywhere. This is because we rescale text images to a small and fixed size which results in low vertical gradient means in the central region, and the variations of text position cause the large STDs in the top and bottom regions. In addition to the aforementioned differences, the authors of [25] find that the STD of horizontal derivatives are large in the central region because texts have different shapes. However, we find that the STD of both horizontal and vertical derivatives are small in the central regions. This means that the differences of text images are suppressed after resizing, which enables us to use simple features and classifiers with good generalization to unseen language types.

We resize the ground-truth English text images from the ICDAR 2013 Robust Reading Competition training dataset and IIIT 5K-word dataset [44] into a fixed height of 12 pixels with aspect ratio kept. Thereafter, a total of 60,000 12×12 gray scale text images are randomly sampled from the resized ground-truth text images. As shown in Fig. 8, characters do not have to be located in the center of a training image. Some of the positive images may contain only

Algorithm 1 Feature Extraction.

Input: G - image gradient
Output: Y - features
$Y \leftarrow \emptyset$
initialize 12 convolutional filters F ▷ F are initialized to 1/24 or 0
for each i, from 1 to 12 **do**
  $mean \leftarrow$ convolving F_i with G ▷ Gradient mean of the ith rectangle
  $std \leftarrow$ convolving F_i with G^2
  $std \leftarrow sqrt(std - mean^2)$ ▷ Gradient std of the ith rectangle
  $Y \leftarrow$ concatenate $mean$, std
end for
return Y.

Algorithm 2 Feature Concatenation.

Input: G_x - image horizontal gradient, G_y - image vertical gradient
Output: Y - features
$feat_x \leftarrow$ feature extraction G_x
$feat_y \leftarrow$ feature extraction G_y
$Y \leftarrow$ concatenate $feat_x$, $feat_y$
return Y.

Fig. 8. Training images. Left: random sampled positive training images. Right: Negative training images.

part of a character (Row 1, Column 5 in the left figure of Fig. 8 only contains part of letter "m") or different parts from different characters (Row 1, Column 3 in the left figure of Fig. 8). The intent of this is twofold. Firstly, we do not need to manually label characters or digits, since only word level annotation is given. Secondly, as shown in Fig. 9, when using a sliding window approach, the text detector is trained to assign a high score to a window which contains transitions between two adjacent characters. This facilitates scoring of text line candidates which we describe in the next section. Therefore, we characterize our method as *text region detection* rather than character detection in [29,37,38]. Non-text images of the same size are randomly sampled from the background of the ICDAR 2013 training images across different scales. In order to verify the generalization ability of our method to other unseen languages such as Chinese and Korean, our training set only contains the images of 52 characters from the English alphabet (26 uppercase, 26 lowercase letters) and 10 digits.

3.2. Multi-channel text line detection

We choose Random Forest [45–47] as our text region detector due to its efficiency and performance. We set the number of decision trees to K, and the maximum depth of a tree is set to 64. During testing, we resize the query image in different scales, and

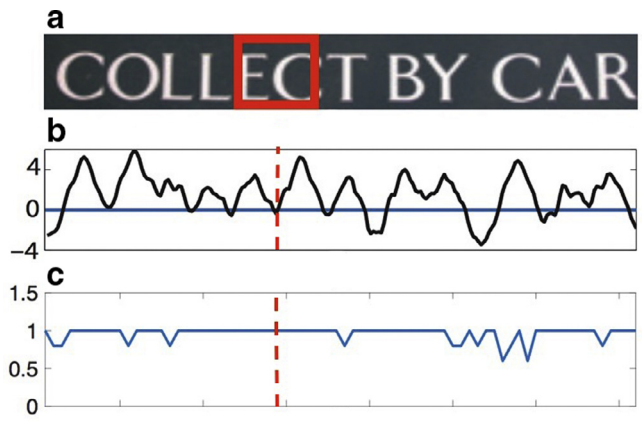

Fig. 9. Text detection responses in a line. (a) Test image. (b) Detector response from [29] which used a CNN for text detection. (c) Detector response of our method where the y-axis indicates the probability of an image window being text region. Our method gives high probability to windows containing transitions between two adjacent characters in contrast to traditional character based method.

use Random Forest to predict the probability of a 12×12 image window being text region, based on the label statistics. Let $p_r(c|x)$ be the empirical distribution of the training data gathered in that terminal node where $c \in \{\text{text}, \text{non-text}\}$ represents the label. An individual decision tree is likely to cause overfitting, but combining the predictions from all trees into a single forest prediction $p(c|x)$ alleviates the issue of overfitting. We choose averaging as the combining method:

$$p(c = \text{text}|x) = \frac{1}{K} \times \sum_{r=1}^{K} p_r(c = \text{text}|x) \tag{3}$$

where x represents the extracted features from an image window and $p_r(c = \text{text}|x)$ represents the prediction of the r-th tree. The averaged prediction $p(c = \text{text}|x)$ is an important cue in scoring text line proposals.

Through extensive experiments, we find that the performance of sliding window method over single gray scale images is limited, due to the loss of color information. Converting an image from RGB color space into gray scale makes text and its corresponding background indistinguishable in severe environments, e.g., non-uniform lighting conditions, as shown in Fig. 10. In the first row of Fig. 10, we observe that the word "Royal" in the C_B and C_R channels have a more distinct appearance than that from the gray scale image. To overcome this drawback, we propose to detect text through three color channels: Gray, C_B and C_R. Note that the C_B and C_R values are in the range of [16,240]. In order to reuse the Random Forest trained on gray scale images whose pixel value is in the range [0,255], we linearly normalize C_B and C_R to the range of [0,255] using Eqs. (4) and (5):

$$I_{CB} = \left(112 - \frac{37.797}{256} \times I_R - \frac{74.203}{256} \times I_G + \frac{112}{256} \times I_B\right) \times \frac{255}{224}, \tag{4}$$

$$I_{CR} = \left(112 + \frac{112}{256} \times I_R - \frac{93.7860}{256} \times I_G - \frac{18.2140}{256} \times I_B\right) \times \frac{255}{224} \tag{5}$$

where I_R, I_G and I_B represent the intensity values in the *RGB* color space and I_{CB} and I_{CR} represent the normalized C_B and C_R intensity values. After linear scaling, we can directly use the text region detector to scan over the normalized I_{CB} and I_{CR} image channels to get response maps as shown in Fig. 10.

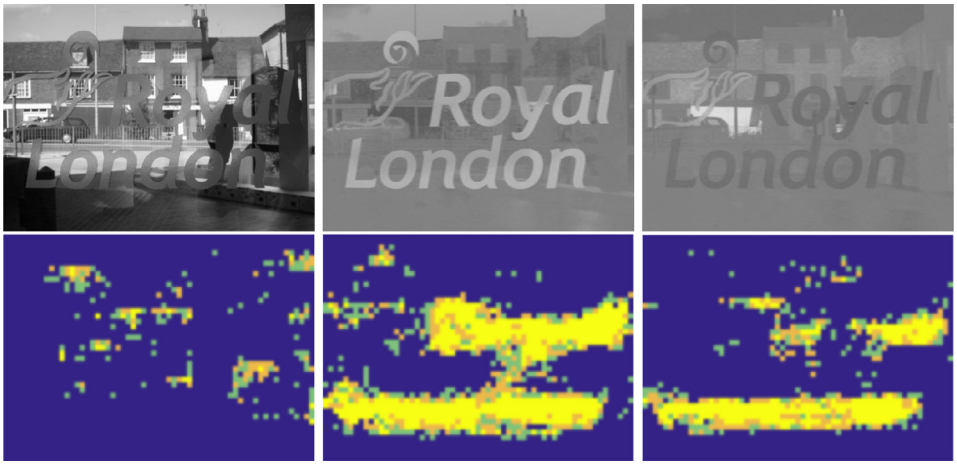

Fig. 10. First row: example image in the Gray, C_B and C_R image channels. Second row: probability maps, where yellow represent probability measure close to 1. For simplicity, we only show the procedure at a single scale. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.3. Text line proposal generation and scoring

Given the response map $R(x, y)$ of image channels, we firstly obtain text line candidates by merging together windows which are likely to be text regions in the horizontal direction, then we assign new scores to each of the text line candidates. Details of the text line candidate generation is shown in Algorithm 3. The computational complexity of text line candidates generation Algorithm 3 is $O(N)$, where N is the number of pixels in images.

Algorithm 3 Text line candidates generation.

Input: $R \in \mathcal{R}^2$ - response map from a given image channel
Output: Sets of text line level candidates L and their scores
$L \leftarrow \emptyset$
for each x, $1 \le x \le$ height of R, from first row to last row **do**
  $L_x \leftarrow \emptyset$
  for each y, $1 \le y \le$ width of R, from first column to last column **do**
    if $R(x, y) \ge 0.5$ **then**
      $l \leftarrow L_x(end)$ get the last element of L_x
      $RMP \leftarrow$ coordinates of the right-most point in l
      if $dist((x, y), RMP) \le t$ (Equation 6) **then**
        $l \leftarrow l \cup \{(x, y)\}$
        break
      end if
      if $R(x, y)$ is not merged into any $l \in L_x$ **then**
        $nc \leftarrow \{(x, y)\}$ ▷ start a new text line candidate
        $L_x \leftarrow L_x \cup \{nc\}$
      end if
    end if
  end for
  $L \leftarrow L \cup L_x$
end for
Extract and score bounding boxes for each text line candidate in L.

We first introduce some notations. Let $R(x, y_1)$ and $R(x, y_2)$ denote the responses of two image windows (x, y_1, w, h) and (x, y_2, w, h), where (x, y_1) and (x, y_2) are the top-left corners, h represents height and w is width respectively, and we set $w = h = 12$. Define $L_x = \bigcup_{i=1} l_i$ where each l_i is a set of top-left corners of windows which can be merged into a single text line candidate in the xth row of $R(x, y)$. The windows are merged according to the following rule. Suppose $(x, y_1) \in l_i$ and (x, y_1) is the right-most point in l_i. Let $R(x, y_2)$ be the nearest neighbour whose response is greater or equal to 0.5 on the right side of (x, y_1), i.e., $y_2 \ge y_1$, which has not been assigned to any l_i. If the following criterion is satisfied, (x, y_2) is merged into l_i.

$$dist((x, y_1), (x, y_2)) = |y_2 - y_1| \leqslant t \tag{6}$$

Eq. (6) says if the horizontal distance of two successive responses $R(x, y_1)$ and $R(x, y_2)$ is smaller or equal to t, then (x, y_2) becomes an element of l_i. This process is repeated until we reach the end of the row. Let (x, y_{min}) and (x, y_{max}) be two top-left corners with the minimum and maximum y in l_i. The bounding box of a text line candidate is given by $(x, y_{min}, y_{max} - y_{min} + w, h)$.

Next, we propose to assign each text line candidate l_i with text score C given by:

$$C(l_i) = \frac{1}{|l_i|} \sum_{(x,y) \in l_i} R(x, y) + (|l_i| - 1) \times \epsilon \tag{7}$$

where $|l_i|$ represents the cardinality of l_i. Eq. (7) consists of two terms, the first term is the average prediction of the grouped windows based on the predictions of the text region detector. The second term is added from the observation that text lines normally have a larger width than height. Thus, the second term of Eq. (7) can be regarded as a reward term for *wide* bounding boxes. If $|l_i| = 1$, i.e., if there is only one window in l_i, the reward term is zero. For instance, bounding box of size 12×48 is likely to contain text lines than bounding box of size 12×12. We have tuned parameter ϵ on the ICDAR 2013 Robust Reading Competition training dataset, where we set $\epsilon = 0.01$.

We compared the recall of our detection scheme with or without the reward term in Fig. 11 (a). We observe that, with the reward term, we obtain higher recall with the same number of proposals. Since our goal is to illustrate the influence of the reward term, we only evaluated the performance on small image scales of the training dataset, which would result in low recall. The experiment justifies that scoring with the reward term on wide regions assigns higher recall with less number of proposals. Fig. 11 (b) shows that text lines of varying widths are successfully selected as proposals.

After obtaining text line candidates and their scores, a maximum of top T scored text line candidates are extracted from a total of $3T$ text line candidates consisting of T candidates from the response maps of each channel. We then apply NMS so as to merge and suppress overlapped text line candidates where we compute the intersection over union (IoU) ratio of text line candidates. The NMS algorithm greedily selects highest scoring text

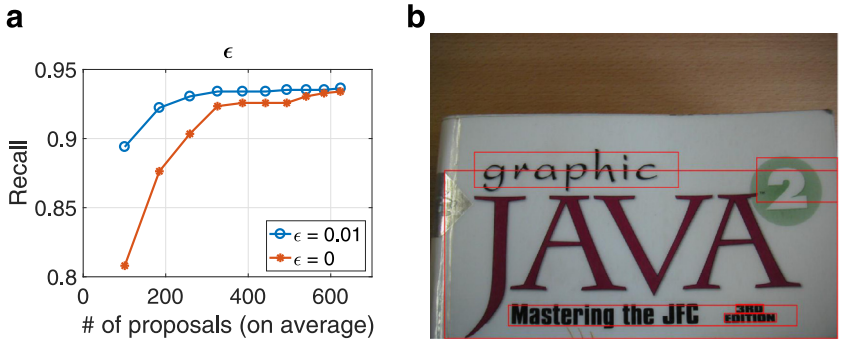

Fig. 11. (a) Performances comparison with and without the reward term. (b) Example of positive proposals (red bounding boxes) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

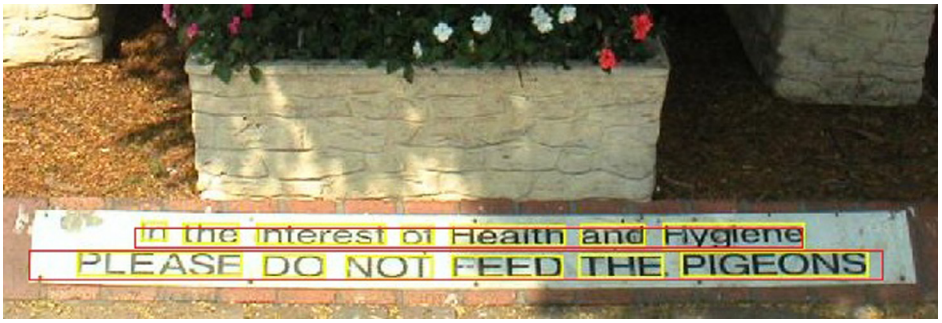

Fig. 12. The IoU ratio of detected bounding boxes (red) with each of the corresponding detected ground-truth bounding box (yellow) is smaller than 0.5. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

line candidates, and deletes or merges less confident overlapped text line candidates to remove duplicated detections since they are likely to cover the same object. In our method, NMS is used as follows: suppose we have two text line candidates bounding boxes $a = (x_1, y_1, w_1, h_1)$ and $b = (x_2, y_2, w_2, h_2)$ and corresponding scores s_1 and s_2, respectively. We assume that $s_1 > s_2$. We first compute the intersection area of the two text line candidates bounding boxes which is defined as

$$\begin{aligned} I(a,b) &= \max(0, [\min(x_1+w_1, x_2+w_2) - \max(x_1, x_2)]) \\ &\quad \times \max(0, [\min(y_1+h_1, y_2+h_2) - \max(y_1, y_2)]), \\ U(a,b) &= w_1 \times h_1 + w_2 \times h_2 - I(a,b). \end{aligned}$$

If the IoU ratio $I(a, b)/U(a, b) \in [0.6, 1)$, a and b are merged into one new bounding box with coordinates given by

$$\begin{aligned} &(\min(x_1, x_2), \min(y_1, y_2), \max(x_1+w_1-1, x_2+w_2-1), \\ &\max(y_1+h_1-1, y_2+h_2-1)) \end{aligned}$$

and scored by s_1; if $I(a, b)/U(a, b) \in [0.5, 0.6)$, a is kept and b is discarded; and if $I(a, b)/U(a, b) \in [0, 0.5)$, both a and b are kept.

3.4. Detection metric

In the object detection, a candidate bounding box is said to be a positive detection if the candidate bounding box has overlapped a sufficient area with a ground-truth bounding box. This occurs when the ratio of IoU is above a certain threshold which is typically set to 0.5. Although 0.5 IoU is satisfactory in object detection, but it is not so for text detection. This is because our method generates bounding boxes at the text line level. For example, as shown in Fig. 12, ground-truth texts (yellow rectangles) in the horizontal direction are enclosed by detected bounding boxes (red rectangles). The IoU of the detected bounding box with each of the ground-truth bounding box at word level is below 0.5. However, most scene text detection methods regard these detected bounding boxes as positive detections. Moreover, only text line level annotations are given in the multilingual datasets. Thus, we propose to use the following criteria for performance evaluation.

Given a set of ground-truth bounding boxes $G_i, i \in \{1, 2, \ldots, |G_i|\}$ and a set of proposals $D_j, j \in \{1, 2, \ldots, |D_j|\}$, area recall σ_{ij} and area precision τ_{ij} are defined as:

$$\sigma_{ij} = \frac{\text{Area}(G_i \cap D_j)}{\text{Area}(G_i)}, \tag{8}$$

$$\tau_{ij} = \frac{\text{Area}(G_i \cap D_j)}{\text{Area}(D_j)} \tag{9}$$

where function Area$(\cdot)$ denotes the area of a given region. We consider two matching criteria: One-to-One matches and Many-to-One matches [15,48]:

One-to-One matches: D_j matches with a ground-truth bounding box G_i, if the following criteria are satisfied:

$$\sigma_{ij} > 0.8, \tag{10}$$

$$\tau_{ij} > 0.4. \tag{11}$$

Many-to-One matches: We say D_j matches a set S_m of ground-truth bounding boxes, if

1. The proportion of detected area of each ground-truth bounding box is at least 0.8, i.e.,

$$\forall G_i \in S_m : \sigma_{ij} \geq 0.8 \tag{12}$$

2. The ratio of text area to area of D_j is at least 0.4, i.e.,

$$\sum_{i \in S_m} \tau_{ij} \geq 0.4 \tag{13}$$

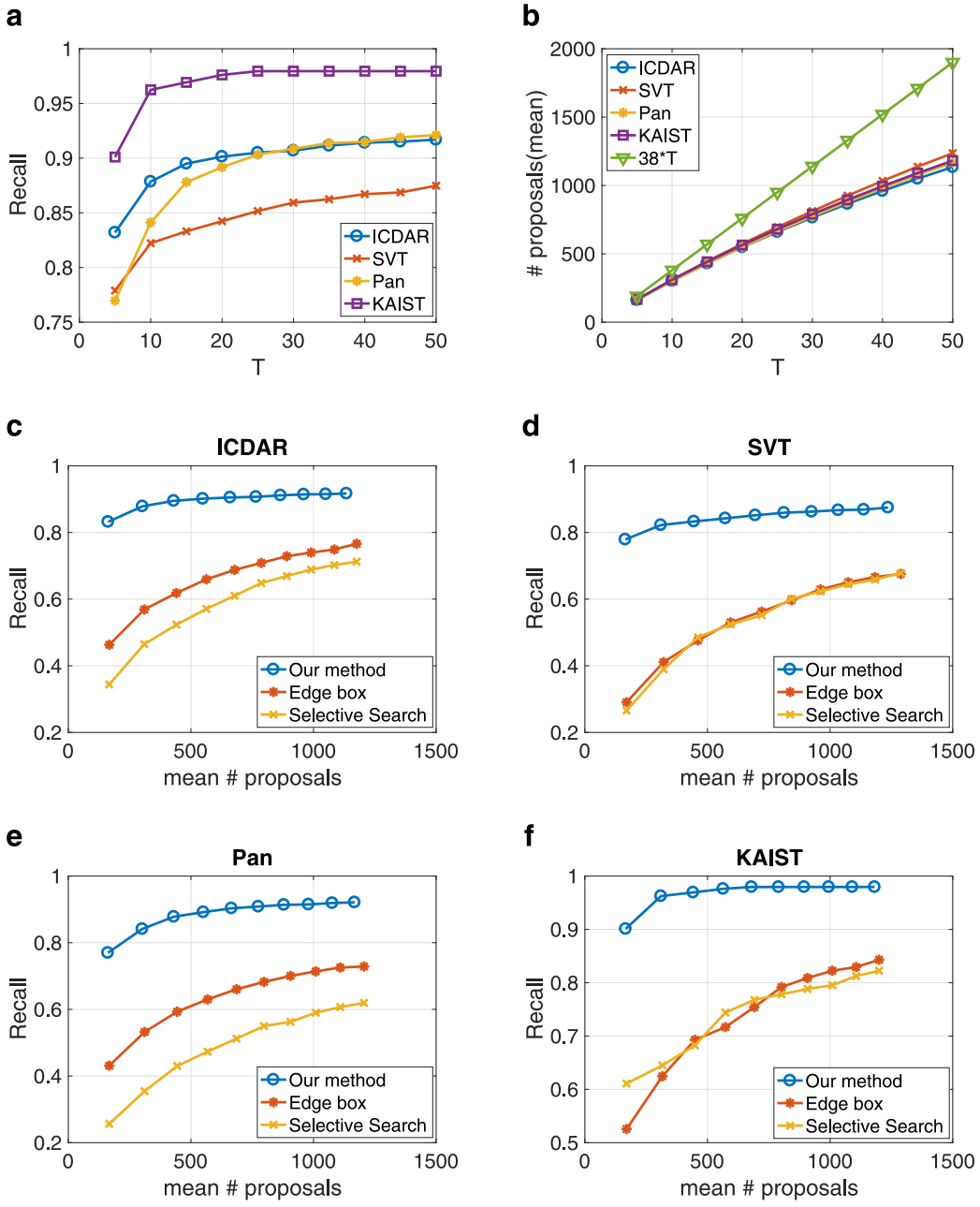

Fig. 13. Experimental results on different datasets. (a) Recall for different values of T which is the threshold on the maximum number of bounding boxes which can be extracted as proposals at each scale. (b) Average number of proposals for different T per image. (c) Comparison on the ICDAR 2013 dataset. (d) Comparison on the the SVT dataset. (e) Comparison on the Pan's multilingual dataset. (f) Comparison on the randomly sampled images from the KAIST Scene Text dataset.

3. Every G_i in S_m must satisfy a height constraint:

$$\forall G_i \in S_m : \frac{\max(h(D_j), h(G_i))}{\min(h(D_j), h(G_i))} \leq 1.5 \tag{14}$$

where $h(\cdot)$ represents the height of a rectangle bounding box. The height constraint was not considered in [15,48]. We added the additional height constraint to ensure single text line detection. For instance, Eq. (14) can be used to prevent a bounding box which enclose two words "Raffee" and "Pads" of the first image in Fig. 2 from being considered as a positive detection.

In this paper, we do not take One-to-Many matches and Many-to-Many matches into consideration, because multiple detected bounding boxes need to be further grouped into word level bounding boxes. If a detected bounding box satisfies either One-to-One matches or Many-to-One matches, we consider it as a positive detection.

4. Experimental results

In this section, we evaluate our method on four public databases. We first evaluate our method on two widely used English text datasets: the ICDAR 2013 Robust Reading Competition dataset (Challenge 2: Reading Text in Scene Images) and the SVT dataset. Next we evaluate the performance of our method on two unseen multilingual datasets: Pan's multilingual image dataset (Chinese and English), and randomly sampled images from KAIST Scene Text dataset (English and Korean).

All the testing images (after smoothing using the Guided Filter [49]) are resized to a fixed height of 800 pixels with the aspect

Fig. 14. Qualitative examples of text line proposals on the ICDAR 2013 testing images.

ratio kept. A fixed 12×12 sliding window is then applied to scan over images of 38 quantized scales with height spaced between 800 and 12 pixels. From each scale, a maximum of T top scored text line candidates are selected as proposals where T is chosen from $\{5, 10, 15, \ldots, 50\}$. We use the recall as the evaluation criterion defined as:

$$\text{Recall} = \frac{(\text{Number of true positives})}{(\text{Number of true positives and false negatives})} \quad (15)$$

where a true positive is a correct detection (One-to-One Matches or Many-to-One Matches) of a ground-truth text, and a false negative is an incorrect negative classification. We compare our method with Edge Boxes and Selective Search on four datasets under the same detection criteria. The proposed method is implemented in Matlab. All experiments are carried out on a laptop computer with 2.9 GHz Intel i7 CPU and 16Gb RAM.

4.1. Recall-proposal evaluation and performance comparison

In this section, we report the recall-proposal evaluation by testing our method on four widely used datasets. The proposed method is trained on the gray scale training set described in Section 3.1 which consists of only English characters and digits. Using 5 fold cross-validation, we set the number of trees K to 5. The horizontal distance threshold t in Eq. (6) is set to 17.

ICDAR 2013 robust reading dataset: The ICDAR 2013 Robust Reading Competition (challenge 2: Reading Text in Scene Images) dataset includes 229 training images and 233 testing images. The dataset contains a variety of texts from urban scene images which makes it popular for testing text detection algorithms.

Street view text dataset: There are 350 natural scene images and a total of 725 labelled English texts in the SVT dataset. The SVT dataset was obtained from Google street view, hence the images in this dataset are noisy and have a low resolution. From the

Fig. 15. Examples of the true positive proposals for the SVT test dataset.

Fig. 16. Examples of the true positive proposals for Pan's testing images.

Fig. 17. Examples of the true positive proposals for sampled images from KAIST scene text dataset.

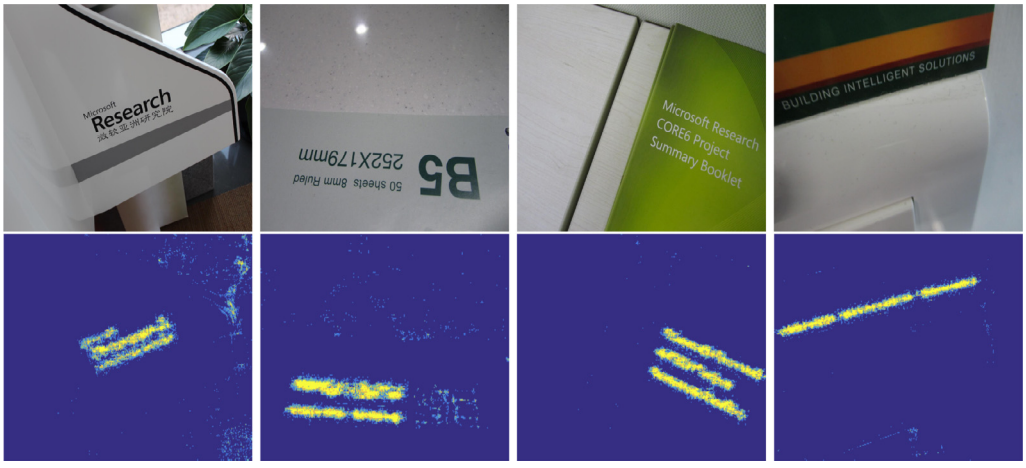

Fig. 18. Response maps on texts of different orientations. For simplicity, we only show the response map at a single scale.

detection point of view, the SVT dataset is more challenging than the ICDAR 2013 Robust Reading Dataset, and thus is a good benchmark for testing robustness in handling low quality and noisy images. We directly apply our method to 250 testing images from the SVT dataset in order to evaluate the robustness of our method.

Pan's multilingual dataset: Pan's multilingual image dataset contains both English and Chinese texts. There are 248 training images and 239 testing images in this dataset consisting of outdoor scene images under different illuminations with texts of different fonts and scales. We test our method on the 239 testing images of Pan's multilingual image dataset so as to verify the generalization ability to unseen languages without retaining our method on this dataset.

Sampled KAIST scene text dataset: The KAIST scene text dataset contains images from natural scene environments. The dataset is categorized into Korean, English, and Mixed (Korean, English). We randomly sampled 118 images from the KAIST scene text dataset, and evaluated our method on the sampled testing images based on the text line level annotations.

Fig. 13(a) shows the recall with varying parameter T on the aforementioned datasets. When T is set to 5, our method obtains 83.2%, 77.9%, 76.9% and 90.1% recall on the ICDAR 2013 dataset, SVT dataset, Pan's Multilingual Dataset and sampled KAIST dataset, respectively, with an average of 170 proposals. From the results we conclude that the proposed features have good discriminative ability, and that our scoring scheme which assigns higher scores to text line candidates is effective. When T is increased to 50, our method obtains up to 91.69% recall on the ICDAR 2013 dataset with 1135 proposals, 87.48% recall on the SVT dataset with 1237 proposals, 92.11% recall on the Pan's Multilingual Dataset with 1168 proposals and 97.95% recall on the sampled KAIST dataset with an average of 680 proposals. The experimental results confirm that, even though our method is trained on a different training set, it is robust to noisy images and has good generalization ability to unseen languages. Text proposals closest to each ground truth bounding box are shown in Figs. 14–17.

Fig. 13(b) shows the average number of proposals per image for different values of T. We also draw $38 \times T$ which is the upper bound of the number of proposals for different T, where 38 is the number of image scales. The mean number of proposal is smaller than $38 \times T$, because for some image scales, the number of proposals that can be extracted, e.g., from an 18×18 image, is smaller than some T.

We also compare the performance of our method with Edge Boxes and Selective Search. For Selective Search, we use the quality

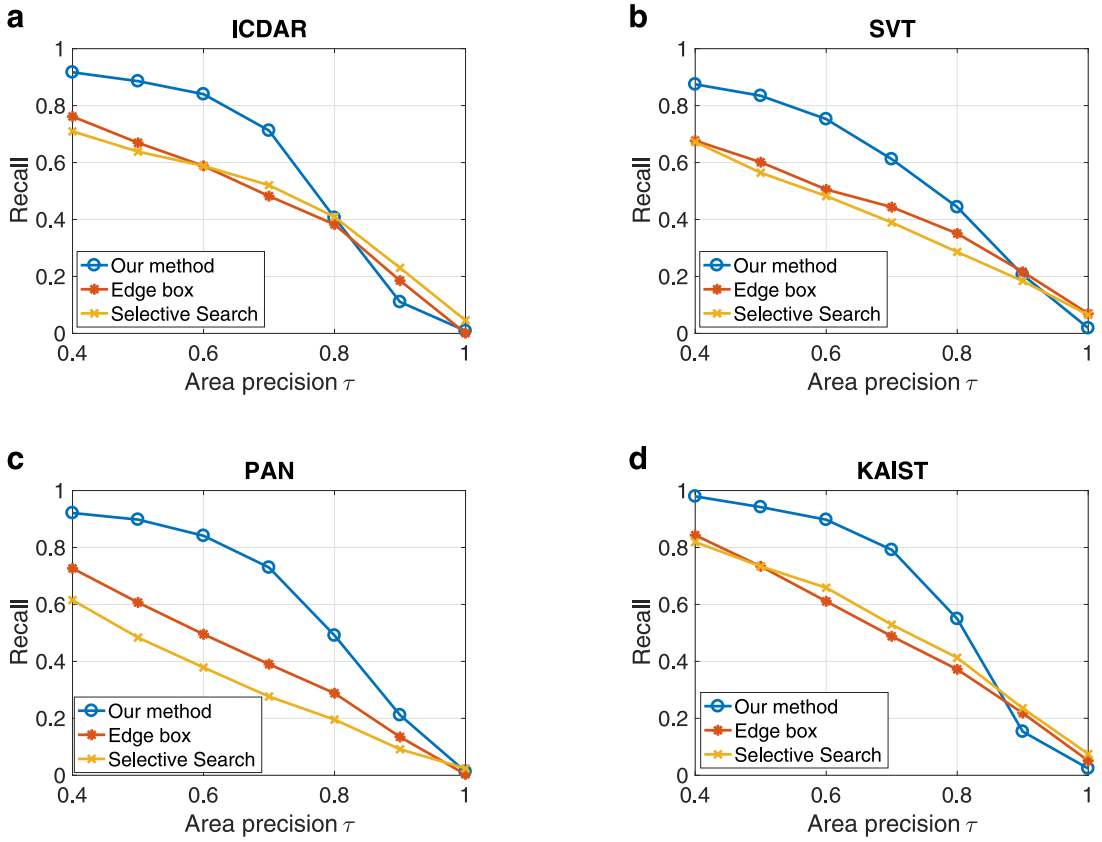

Fig. 19. Proposal comparison of our method to Edge Boxes and Selective Search under the conditions $T = 50$ and $\sigma = 0.8$.

mode and the minimum width of bounding boxes is set to 12. For Edge Boxes, we use the default parameters as suggested in [13].

Fig. 13(c)–(f) show the performance comparisons on the ICDAR 2013 dataset, the SVT dataset, Pan's Multilingual dataset and sampled KAIST scene text dataset, respectively. As shown in these figures, we obtain higher recall than that of Edge Box and Selective Search. For instance, our system obtains over 83.2% recall with 165 text proposals (on average) on the ICDAR testing dataset. By contrast, Edge Box and Selective Search only achieves recall of 46.3% and 34.4%. With an average of 1135 proposals, our method obtains a best recall of 91.69%, Edge Boxes achieves 76.6% recall and Selective Search achieves 71.2% recall. We can find similar results on the noisy SVT dataset and the two multilingual datasets.

We can also limit the number of candidate bounding boxes since every candidate bounding box is given a score by keeping the top scored candidate bounding boxes from the proposal pool when $T = 50$. As shown in Fig. 20, our method achieves high recall even with low hundreds proposals per image, e.g., our method achieves 79.2%, 72.5%, 67.1% and 82.3% recall rate by keeping the top 50 scored proposals on the IDCAR 2013, SVT, Pan's Multilingual Dataset and sampled KAIST dataset. Moreover, with a few hundreds number of proposals, our method is able to achieve over 80% recall rate.

Fig. 18 depicts some examples of the application of our method to texts of different orientations. We observe that, even though our method is trained to detect horizontal text lines, it also generates high responses to slightly oblique texts. We believe this is because the training images are not perfectly horizontal, which causes our method to be trained on, to some extent, text lines rotated by small degrees. This opens possibility of our method to be extended to detect multi-oriented text lines. Such extension is part of our ongoing work.

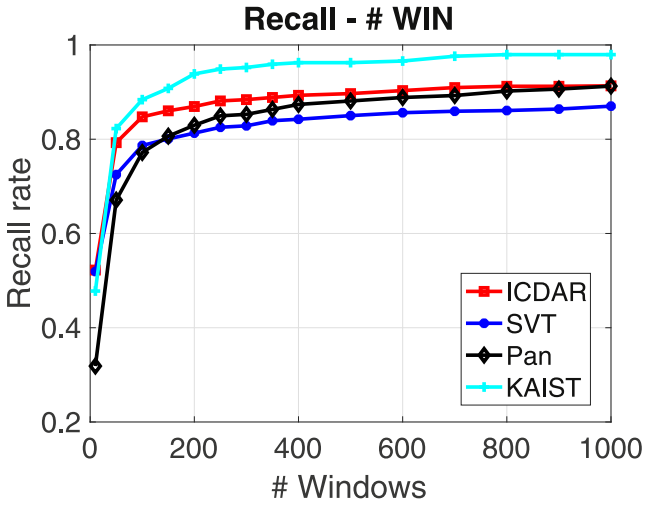

Fig. 20. Recall rate when the number of windows (per image) are limit to a specific number on various datasets. Our method achieves over 80% recall rate on the 4 datasets using the top scored 200 proposals.

4.2. Proposal quality evaluation

In this section, we evaluate the recall by varying the threshold of area precision τ of the One-to-One matches and the Many-to-One matches under the condition that σ is fixed to 0.8. Fig. 19 shows the recall versus the area precision τ of our method, Edge Boxes and Selective Search with $T = 50$ and $\tau = 0.8$. Our method outperforms Edge Boxes and Selective Search over a wide

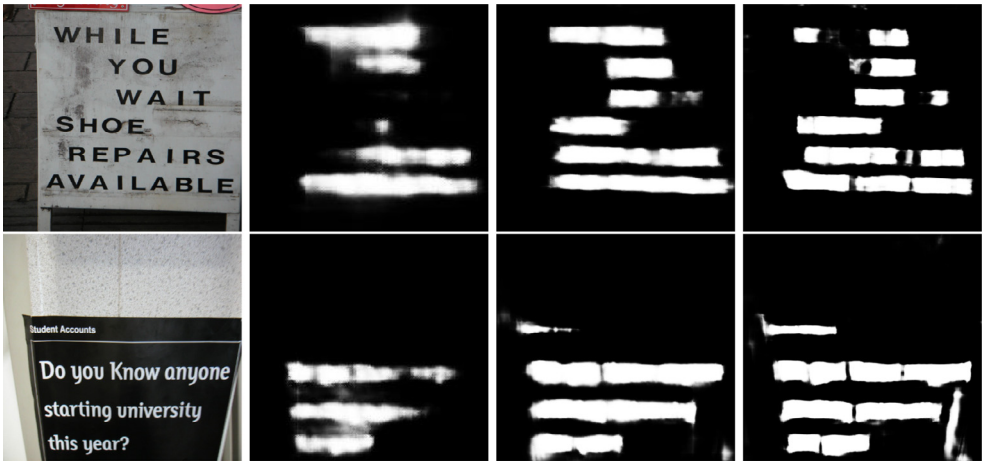

Fig. 21. Test images and Text-Block FCN outputs. From left to right, test image, response maps of which the heights are 200, 500, 1,000 pixels, respectively.

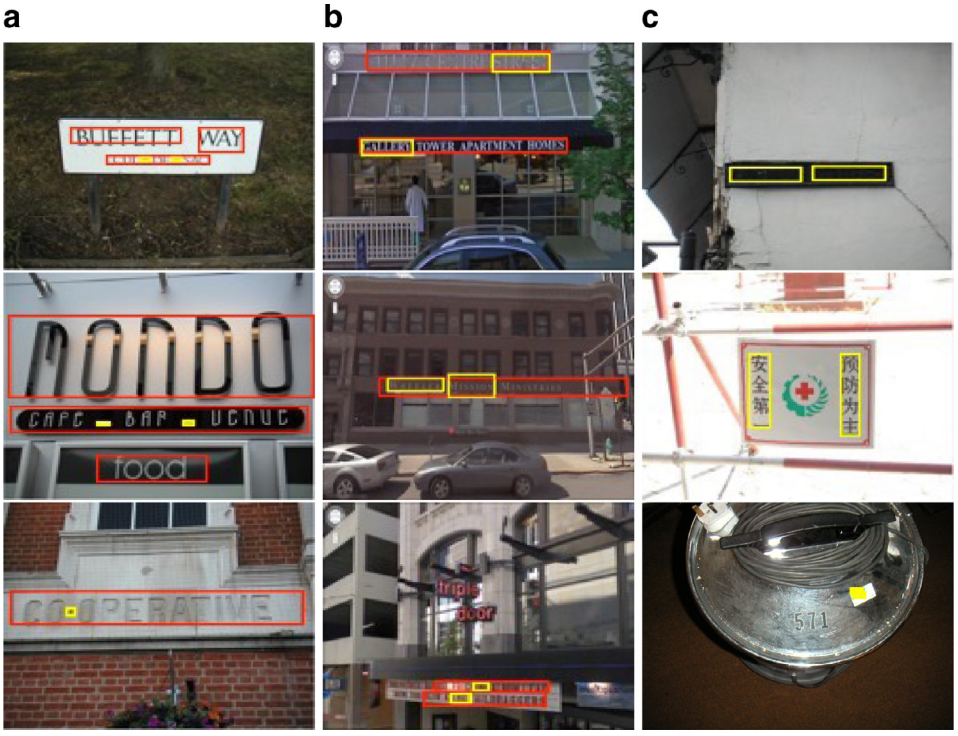

Fig. 22. Failed examples. Yellow rectangles represent ground-truth texts which are failed to be detected by our method, red rectangles represent detected bounding boxes. The first two columns show some misclassified ground-truth texts. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

range of area precision which are desired in practice with τ between 0.4 to 0.8.

Next, we comment on the computational overhead of the proposed method. Our method requires an average of 30 s for one image. Because our method is implemented in Matlab and is not optimized for speed, there is room for improvements in processing time. Most of the computational time is spent on the feature extraction, e.g., it takes 7 s to extract features from an 800×1200 images and 5 s from next image scale of size 715×1072, and the time for the feature extraction of the first two image scales accounts for 40% of the overall computational time. Since the proposed features are simple and can be computed by convolving 12 mean filters over the image gradients, the computational time can be greatly reduced by using efficient programming languages and GPU computing.

4.3. Comparison with fully convolutional networks (FCN)

We compare our method with FCN based text detection method [50], which has shown good performances for generating text line proposals. We have fully implemented the Text-Block FCN with PyTorch[1], and compared our method with [50]. Specifically, we compared our results with the outputs generated at different steps as specified in [50] such that

- Text-Block FCN (step 1)
- Text-Block FCN + MSER (step 1 and step 2)

in terms of recall rate and number of proposals. Our implementation is identical to [50] except the following modifications:

[1] https://github.com/pytorch/pytorch.

Table 1
Text proposal performance comparisons: recall rate (mean # proposals)

Method	ICDAR2015	SVT	Pan	KAIST
Text-Block FCN	0.47 (17)	0.31 (22)	0.33 (18)	0.41 (24)
Text-Block FCN + MSER	0.68 (97)	0.32 (118)	0.51 (120)	0.39 (118)
Our method (T = 50)	0.65 (20)	0.65 (20)	0.46 (20)	0.64 (20)
Our method (T = 50)	0.85 (100)	0.79 (100)	0.77 (100)	0.88(100)
Our method (T = 50)	0.90 (600)	0.86 (600)	0.89 (600)	0.97 (600)

1. *Training data.* 11,450 500×500 image patches are randomly sampled from the ICDAR 2013 Robust Reading Training Dataset.
2. *Learning rate.* The learning rate is set to constant 0.001 during the training stage, and the training stops at 100 epoch.
3. *Orientation estimation.* Since the testing datasets only contain horizontal texts, the orientation estimation is set to 0 (horizontal text lines).

The comparison results are shown in Table 1. From Table 1, we observe that Text-Block FCN generates coarse bounding boxes, which results in low recall at the word (One-to-One Matches) or text line level (Many-to-One Matches). Thus, the detected text-blocks need to be further processed so that it is split into text lines, as in Text-Block FCN+MSER in our experiment. As shown in Table 1, our method outperforms both Text-Block FCN and Text-Block FCN + MSER on all test datasets. In addition, Text-Block FCN does not have good generalization ability to Chinese and Korean when it is trained only on English. Meanwhile, our method achieves higher recall with the same number of proposals.

4.4. Limitation of the proposed method

Although our method achieves good performance on public datasets, it has the following limitations. Our method utilizes the sliding window based method, and thus it inherits some limitations from the method. Several failed examples are shown in Fig. 22. In Fig. 22 (a), Ground-truth texts from ICDAR 2013 dataset are marked as misses due to the height constraint (Eq. (14)). In Fig. 22 (b), unlabelled texts and ground-truth texts from SVT failed to be detected as Many-to-One matches due to the lack of annotations. Our method can fail to detect very low contrast texts, non-horizontal texts and non-recognizable texts as shown in Fig. 22 (c).

5. Conclusion and future work

In this paper, we present a simple and effective language-independent scene text line proposal generation method. Our method aims at detecting text achieving high recall with only a few thousands of high quality proposals. We believe our method is useful in text detection tasks, because it can greatly reduce the number of windows at the cost of losing a few correct detections. Also the proposed method for feature extraction is simple and computationally efficient. The experiments on four different datasets demonstrate that the proposed method not only is able to effectively generate a small set of candidate bounding boxes from a large set of possible bounding boxes while maintaining high recall, but also exhibits strong robustness to noisy images and different language types. Our future work includes improving the recall, further reducing the number of proposals, and extending our method to multi-oriented text detection.

Acknowledgment

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2018R1A2B6007130) and (No. 2016R1A2B1014934), and in part by the Korean MSIT (Ministry of Science and ICT), under the National Program for Excellence in SW (2015-0-00936) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

[1] D. Zheng, Y. Zhao, J. Wang, An efficient method of license plate location, Pattern Recognit. Lett. 26 (15) (2005) 2431–2438.
[2] E.R. Lee, P.K. Kim, H.J. Kim, Automatic recognition of a car license plate using color image processing, in: Proceedings of the IEEE International Conference Image Processing, ICIP, Vol. 2, IEEE, 1994, pp. 301–305.
[3] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in: Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.
[4] I. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, V. Shet, Multi-digit number recognition from street view imagery using deep convolutional neural networks, in: Proceedings of International Conference on Learning Representation, 2014.
[5] A. De La Escalera, L.E. Moreno, M.A. Salichs, J.M. Armingol, Road traffic sign detection and classification, IEEE Trans. Ind. Electron. 44 (6) (1997) 848–859.
[6] P. Sermanet, S. Chintala, Y. LeCun, Convolutional neural networks applied to house numbers digit classification, in: Proceedings of the Twenty First International Conference on Pattern Recognition (ICPR), IEEE, 2012, pp. 3288–3291.
[7] Q. Ye, D. Doermann, Text detection and recognition in imagery: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 37 (7) (2015) 1480–1500.
[8] J.R. Uijlings, K.E. van de Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.
[9] M.-M. Cheng, Z. Zhang, W.-Y. Lin, P. Torr, Bing: binarized normed gradients for objectness estimation at 300fps, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3286–3293.
[10] C.L. Zitnick, P. Dollár, Edge boxes: locating object proposals from edges, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 391–405.
[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR, IEEE, 2009, pp. 248–255.
[12] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[13] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, Int. J. Comput. Vis. 116 (1) (2016) 1–20.
[14] P. Dollár, R. Appel, S. Belongie, P. Perona, Fast feature pyramids for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 36 (8) (2014) 1532–1545.
[15] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L.G. i Bigorda, S.R. Mestre, J. Mas, D.F. Mota, J.A. Almazan, L.P. de las Heras, ICDAR robust reading competition, in: Proceedings of the Twelfth International Conference on Document Analysis and Recognition, IEEE, 2013, pp. 1484–1493.
[16] K. Wang, S. Belongie, Word spotting in the wild, in: Proceedings of the European Conference on Computer Vision, Springer, 2010, pp. 591–604.
[17] Y.-F. Pan, X. Hou, C.-L. Liu, A hybrid approach to detect and localize texts in natural scene images, IEEE Trans. Image Process. 20 (3) (2011) 800–813.
[18] J. Jung, S. Lee, M.S. Cho, J.H. Kim, Touch TT: scene text extractor using touchscreen interface, ETRI J. 33 (1) (2011) 78–88.
[19] K. Jung, K.I. Kim, A.K. Jain, Text information extraction in images and video: a survey, Pattern Recognit. 37 (5) (2004) 977–997.
[20] H. Zhang, K. Zhao, Y.-Z. Song, J. Guo, Text extraction from natural scene image: a survey, Neurocomputing 122 (2013) 310–323.
[21] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2963–2970.
[22] L. Neumann, J. Matas, A method for text localization and recognition in real-world images, in: Proceedings of the Asian Conference on Computer Vision, Springer, 2010, pp. 770–783.
[23] L. Neumann, J. Matas, Real-time scene text localization and recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 3538–3545.
[24] X.-C. Yin, X. Yin, K. Huang, H.-W. Hao, Robust text detection in natural scene images, IEEE Trans. Pattern Anal. Mach. Intell. 36 (5) (2014) 970–983.
[25] X. Chen, A.L. Yuille, Detecting and reading text in natural scenes, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR, Vol. 2, IEEE, 2004, pp. II–366.
[26] Y.-F. Pan, X. Hou, C.-L. Liu, Text localization in natural scene images based on conditional random field, in: Proceedings of the Tenth International Conference on Document Analysis and Recognition, IEEE, 2009, pp. 6–10.
[27] K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in: Proceedings of the International Conference on Computer Vision, IEEE, 2011, pp. 1457–1464.
[28] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D.J. Wu, A.Y. Ng, Text detection and character recognition in scene images with unsupervised feature learning, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2011, pp. 440–445.
[29] T. Wang, D.J. Wu, A. Coates, A.Y. Ng, End-to-end text recognition with convolutional neural networks, in: Proceedings of the Twenty First International Conference on Pattern Recognition (ICPR), IEEE, 2012, pp. 3304–3308.
[30] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image and Vis. Comput. 22 (10) (2004) 761–767.

[31] Y. Freund, R.E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of the European Conference on Computational Learning Theory, Springer, 1995, pp. 23–37.
[32] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2013.
[33] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, Vol. 1, IEEE, 2005, pp. 886–893.
[34] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[35] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[36] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[37] M. Jaderberg, A. Vedaldi, A. Zisserman, Deep features for text spotting, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 512–528.
[38] T. He, W. Huang, Y. Qiao, J. Yao, Text-attentional convolutional neural network for scene text detection, IEEE Trans. Image Process. 25 (6) (2016) 2529–2541.
[39] J. Hosang, R. Benenson, P. Dollár, B. Schiele, What makes for effective detection proposals? IEEE Trans. Pattern Anal. Mach. Intell. 38 (4) (2016) 814–830.
[40] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[41] R. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[42] B. Alexe, T. Deselaers, V. Ferrari, Measuring the objectness of image windows, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2189–2202.
[43] B. Alexe, T. Deselaers, V. Ferrari, What is an object? in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 73–80.
[44] A. Mishra, K. Alahari, C. Jawahar, Scene text recognition using higher order language priors, in: BMVC 2012-23rd British Machine Vision Conference, BMVA, 2012.
[45] A. Criminisi, J. Shotton, E. Konukoglu, Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning, Found. Trends® Comput. Graph. Vis. 7 (2–3) (2012) 81–227.
[46] T.K. Ho, Random decision forests, in: Proceedings of the Third International Conference on Document Analysis and Recognition, 1995, Vol. 1, IEEE, 1995, pp. 278–282.
[47] P. Dollár, Piotrs computer vision matlab toolbox, http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html(2014).
[48] C. Wolf, J.-M. Jolion, Object count/area graphs for the evaluation of object detection and segmentation algorithms, Int. J. Doc. Anal. Recognit. (IJDAR) 8 (4) (2006) 280–296.
[49] K. He, J. Sun, X. Tang, Guided image filtering, IEEE Trans. Pattern Anal. Mach. Intell. 35 (6) (2013) 1397–1409.
[50] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, X. Bai, Multi-oriented text detection with fully convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4159–4167.

Kun Fan received his B.S. degree in Communication Engineering from Harbin institute of technology, China in 2010. He is currently working toward the Ph.D. degree in Electrical and Computer Engineering in Korea University. His research interest includes machine learning and computer vision.

Seung Jun Baek received his B.S. degree from Seoul National University in 1998, and M.S. and Ph.D. degrees from the University of Texas at Austin in 2002 and 2007, respectively, in electrical and computer engineering. From 2007 to 2009, he was a Member of Technical Staff with DSP Systems R&D Center, Texas Instruments. In 2009, he joined the College of Information and Communications, Korea University, Korea, where he is currently an associate professor. His research interests include information systems and communication networks, machine learning, compressive sensing, and game theory.