Published at the 25th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, June 16-21, Providence, RI, USA

Real-Time Scene Text Localization and Recognition

Lukáš Neumann Jiří Matas
Centre for Machine Perception, Department of Cybernetics
Czech Technical University, Prague, Czech Republic
neumalu1@cmp.felk.cvut.cz, matas@cmp.felk.cvut.cz

Abstract

An end-to-end real-time scene text localization and recognition method is presented. The real-time performance is achieved by posing the character detection problem as an efficient sequential selection from the set of Extremal Regions (ERs). The ER detector is robust to blur, illumination, color and texture variation and handles low-contrast text.

In the first classification stage, the probability of each ER being a character is estimated using novel features calculated with $O(1)$ complexity per region tested. Only ERs with locally maximal probability are selected for the second stage, where the classification is improved using more computationally expensive features. A highly efficient exhaustive search with feedback loops is then applied to group ERs into words and to select the most probable character segmentation. Finally, text is recognized in an OCR stage trained using synthetic fonts.

The method was evaluated on two public datasets. On the ICDAR 2011 dataset, the method achieves state-of-the-art text localization results amongst published methods and it is the first one to report results for end-to-end text recognition. On the more challenging Street View Text dataset, the method achieves state-of-the-art recall. The robustness of the proposed method against noise and low contrast of characters is demonstrated by "false positives" caused by detected watermark text in the dataset.

1. Introduction

Text localization and recognition in real-world (scene) images is an open problem which has been receiving significant attention since it is a critical component in a number of computer vision applications like searching images by their textual content, reading labels on businesses in map applications (e.g. Google Street View) or assisting visually impaired. Several contests have been held in the past years [10, 9, 20] and the winning method in the most recent ICDAR 2011 contest was able to localize only 62% words correctly [20] despite the fact that the dataset is not fully realistic (words are horizontal only, they occupy a significant part of the image, there is no perspective distortion or significant noise).

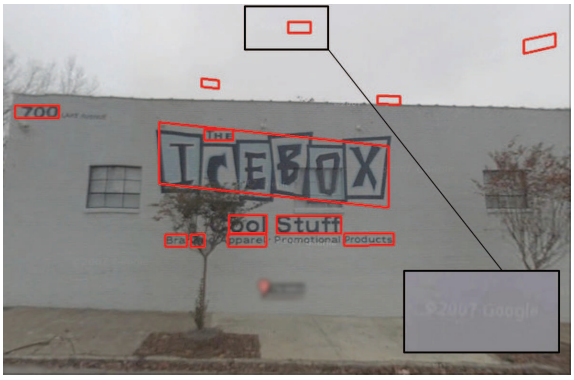


Figure 1. Text detection in the SVT dataset. All "false positives" in the image are caused by watermarks embedded into the dataset. This demonstrates robustness of the proposed method against noise and low contrast of characters (in the bottom-right corner the area of interest is enlarged and contrast artificially increased, "©2007 Google" is readable).

Localizing text in an image is potentially a computationally very expensive task as generally any of the 2^N subsets can correspond to text (where N is the number of pixels). Text localization methods deal with this problem in two different ways.

Methods based on a sliding window [6, 2, 7] limit the search to a subset of image rectangles. This reduces the number of subsets checked for the presence of text to cN where c is a constant that varies between very small values (< 1) for single-scale single-rotation methods to relatively large values ($\gg 1$) for methods that handle text with a different scale, aspect, rotation, skew, etc.

Methods in the second group [5, 17, 14, 15, 24] find individual characters by grouping pixels into regions using connected component analysis assuming that pixels belonging to the same character have similar properties. Connected component methods differ in the properties used (color, stroke-width, etc.). The advantage of the connected component methods is that their complexity typically does not depend on the properties of the text (range of scales, orientations, fonts) and that they also provide a segmentation

which can be exploited in the OCR step. Their disadvantage is a sensitivity to clutter and occlusions that change connected component structure.

In this paper, we present an end-to-end real-time[1] text localization and recognition method which achieves state-of-the-art results on standard datasets. The real-time performance is achieved by posing the character detection problem as an efficient sequential selection from the set of Extremal Regions (ERs). The ER detector is robust against blur, low contrast and illumination, color and texture variation[2]. Its complexity is $O(2pN)$, where p denotes number of channels (projections) used.

In the first stage of the classification, the probability of each ER being a character is estimated using novel features calculated with $O(1)$ complexity and only ERs with locally maximal probability are selected for the second stage, where the classification is improved using more computationally expensive features. A highly efficient exhaustive search with feedback loops (adapted from [15]) is then applied to group ERs into words and select the most probable character segmentation.

Additionally, a novel gradient magnitude projection which allows edges to induce ERs is introduced. It is further demonstrated that by inclusion of the gradient projection 94.8% of characters are detected by the ER detector.

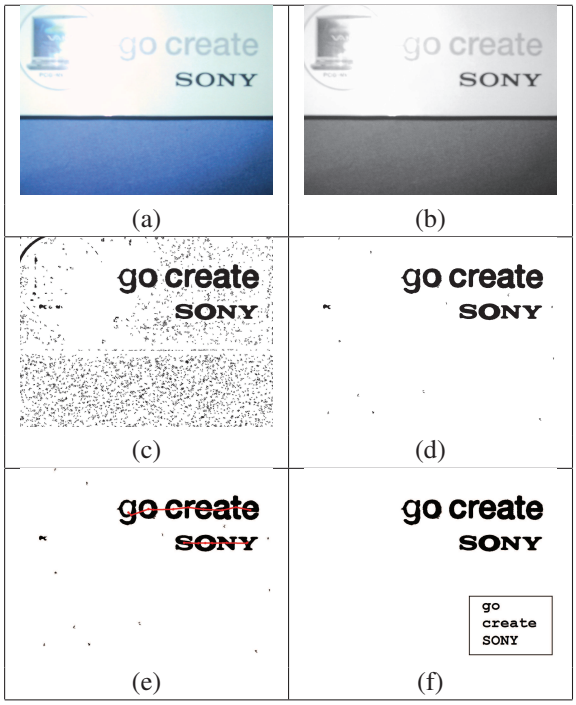
The rest of the document is structured as follows: In Section 2, an overview of previously published methods is given. Section 3 describes the proposed method. In Section 4, the experimental evaluation is presented. The paper is concluded in Section 5.

2. Previous Work

Numerous methods which focus solely on text localization in real-world images have been published [6, 2, 7, 17]. The method of Epstein et al. in [5] converts an input image to a greyscale space and uses Canny detector [1] to find edges. Pairs of parallel edges are then used to calculate stroke width for each pixel and pixels with similar stroke width are grouped together into characters. The method is sensitive to noise and blurry images because it is dependent on a successful edge detection and it provides only single segmentation for each character which not necessarily might be the best one for an OCR module. A similar edge-based approach with different connected component algorithm is presented in [24]. A good overview of the methods and their performance can be also found in ICDAR Robust Reading competition results [10, 9, 20].

Only a few methods that perform both text localization and recognition have been published. The method of Wang et al. [21] finds individual characters as visual words using the sliding-window approach and then uses a lexicon to group characters into words. The method is able to cope with noisy data, but its generality is limited as a lexicon of words (which contains at most 500 words in their experiments) has to be supplied for each individual image.

Methods presented in [14, 15] detect characters as Maximally Stable Extremal Regions (MSERs) [11] and perform text recognition using the segmentation obtained by the MSER detector. An MSER is an particular case of an Extremal Region whose size remains virtually unchanged over a range of thresholds. The methods perform well but have problems on blurry images or characters with low contrast. According to the description provided by the ICDAR 2011 Robust Reading competition organizers [20] the winning method is based on MSER detection, but the method



	run time(ms)	No. of ERs
Initial image	-	6×10^6
Classification (1^{st} stage)	1120	2671
Classification (2^{nd} stage)	130	20
Region grouping	20	12
OCR	110	12

(g)

Figure 2. Text localization and recognition overview. (a) Source 2MPx image. (b) Intensity channel extracted. (c) ERs selected in $O(N)$ by the first stage of the sequential classifier. (d) ERs selected by the second stage of the classifier. (e) Text lines found by region grouping. (f) Only ERs in text lines selected and text recognized by an OCR module. (g) Number of ERs at the end of each stage and its duration

[1] We consider a text recognition method real-time if the processing time is comparable with the time it would take a human to read the text.

[2] A www service allowing testing of the method is available at http://cmp.felk.cvut.cz/ neumalu1/TextSpotter

(a) (b) (c)

Figure 3. Intensity gradient magnitude channel ∇. (a) Source image. (b) Projection output. (c) Extremal Regions at threshold $\theta = 24$ (ERs bigger than 30% of the image area excluded for better visualization)

itself had not been not published and it does not perform text recognition.

The proposed method differs from the MSER-based methods [14, 15] in that it tests all ERs (not only the subset of MSERs) while reducing the memory footprint and maintaining the same computational complexity and real-time performance. The idea of dropping the stability requirement of MSERs and selecting a class-specific (not necessarily stable) Extremal Regions was first presented by Zimmermann and Matas [12], who used image moments as features for a monolithic neural network, which was trained for a given set of shapes (e.g. textures, specific characters). In our method, the selection of suitable ERs is carried out in real-time by a sequential classifier on the basis of novel features which are specific for character detection. Moreover, the classifier is trained to output probability and thus extracts several segmentations of a character.

3. The Proposed Method

3.1. Extremal Regions

Let us consider an image $\mathbf{I}$ as a mapping $\mathbf{I} : \mathcal{D} \subset \mathbb{N}^2 \to \mathcal{V}$, where $\mathcal{V}$ typically is $\{0, \ldots, 255\}^3$ (a color image). A channel $\mathbf{C}$ of the image $\mathbf{I}$ is a mapping $\mathbf{C} : \mathcal{D} \to \mathcal{S}$ where $\mathcal{S}$ is a totally ordered set and $f_c : \mathcal{V} \to \mathcal{S}$ is a *projection* of pixel values to a totally ordered set. Let A denote an adjacency (neighborhood) relation $A \subset \mathcal{D} \times \mathcal{D}$. In this paper we consider 4-connected pixels, i.e. pixels with coordinates $(x \pm 1, y)$ and $(x, y \pm 1)$ are adjacent to the pixel (x, y).

Region $\mathcal{R}$ of an image $\mathbf{I}$ (or a channel $\mathbf{C}$) is a contiguous subset of $\mathcal{D}$, i.e. $\forall p_i, p_j \in \mathcal{R}\ \exists p_i, q_1, q_2, \ldots, q_n, p_j : p_i A q_1, q_1 A q_2, \ldots, q_n A p_j$. Outer region boundary $\partial\mathcal{R}$ is a set of pixels adjacent but not belonging to $\mathcal{R}$, i.e. $\partial\mathcal{R} = \{p \in \mathcal{D} \setminus \mathcal{R} : \exists q \in \mathcal{R} : pAq\}$. *Extremal Region* (ER) is a region whose outer boundary pixels have strictly higher values than the region itself, i.e. $\forall p \in \mathcal{R}, q \in \partial\mathcal{R} : \mathbf{C}(q) > \theta \geq \mathbf{C}(p)$, where θ denotes threshold of the Extremal Region.

An ER r at threshold θ is formed as a union of one or more (or none) ERs at threshold $\theta - 1$ and pixels of value θ, i.e. $r = \left(\bigcup u \in R_{\theta-1}\right) \cup \left(\bigcup p \in \mathcal{D} : \mathbf{C}(p) = \theta\right)$, where $R_{\theta-1}$ denotes set of ERs at threshold $\theta - 1$. This induces an *inclusion relation* amongst ERs where a single ER has one or more predecessor ERs (or no predecessor if it contains only pixels of a single value) and exactly one successor ER (the ultimate successor is the ER at threshold 255 which contains all pixels in the image).

Channel	R (%)	P (%)
R	83.3	7.7
G	85.7	10.3
B	85.5	8.9
H	62.0	2.0
S	70.5	4.1
I	85.6	10.1
∇	74.6	6.3

Channel	R (%)	P (%)
I∪H	89.9	6.0
I∪S	90.1	7.2
I∪∇	90.8	8.4
I∪H∪S	92.3	5.5
I∪H∪∇	93.1	6.1
I∪R∪G∪B	90.3	9.2
I∪H∪S∪∇	**93.7**	**5.7**
all (7 ch.)	94.8	7.1

Table 1. Recall (R) and precision (R) of character detection by ER detectors in individual channels and their combinations. The channel combination used in the experiments is in bold

In this paper, we consider RGB and HSI color spaces [3] and additionally an *intensity gradient* channel (∇) where each pixel is assigned the value of "gradient" approximated by maximal intensity difference between the pixel and its neighbors (see Figure 3):

$$\mathbf{C}_\nabla(p) = \max_{q \in \mathcal{D} : pAq} \{|\mathbf{C}_\mathbf{I}(p) - \mathbf{C}_\mathbf{I}(q)|\}$$

An experimental validation shows that up to 85.6% characters are detected as ERs in a single channel and that 94.8% characters are detected if the detection results are combined from all channels (see Table 1). A character is considered as detected if bounding box of the ER matches at least 90% of the area of the bounding box in the ground truth. In the proposed method, the combination of intensity (`I`), intensity gradient (∇), hue (`H`) and saturation (`S`) channels was used as it was experimentally found as the best trade-off between short run time and localization performance.

3.2. Incrementally Computable Descriptors

The key prerequisite for fast classification of ERs is a fast computation of region descriptors that serve as features for the classifier. As proposed by Zimmerman and Matas [12], it is possible to use a particular class of descriptors and exploit the inclusion relation between ERs to incrementally compute descriptor values.

Let $R_{\theta-1}$ denote a set of ERs at threshold $\theta - 1$. An ER $r \in R_\theta$ at threshold θ is formed as a union of pixels of regions at threshold $\theta - 1$ and pixels of value θ, i.e. $r = \left(\bigcup u \in R_{\theta-1}\right) \cup \left(\bigcup p \in \mathcal{D} : \mathbf{C}(p) = \theta\right)$. Let us further assume that descriptors $\phi(u)$ of all ERs at threshold $u \in R_{\theta-1}$ are already known. In order to compute a descriptor $\phi(r)$ of the region $r \in R_\theta$ it is necessary to combine descriptors of regions $u \in R_{\theta-1}$ and pixels $\{p \in \mathcal{D} : \mathbf{C}(p) = \theta\}$ that formed the region r, i.e.

$\phi(r) = \Big(\oplus\phi(u)\Big) \oplus \Big(\oplus\psi(p)\Big)$, where $\oplus$ denotes an operation that combines descriptors of the regions (pixels) and $\psi(p)$ denotes an initialization function that computes the descriptor for given pixel p. We refer to such descriptors where $\psi(p)$ and $\oplus$ exist as *incrementally computable* (see Figure 4).

It is apparent that one can compute descriptors of all ERs simply by sequentially increasing threshold θ from 0 to 255, calculating descriptors ψ for pixels added at threshold θ and reusing the descriptors of regions ϕ at threshold $\theta - 1$. Note that the property implies that it is necessary to only keep descriptors from the previous threshold in the memory and that the ER method has a significantly smaller memory footprint when compared with MSER-based approaches. Moreover if it is assumed that the descriptor computation for a single pixel $\psi(p)$ and the combining operation $\oplus$ has constant time complexity, the resulting complexity of computing descriptors of all ERs in an image of N pixels is $O(N)$, because $\phi(p)$ is computed for each pixel just once and combining function can be evaluated at most N times, because the number of ERs is bounded by the number of pixels in the image.

In this paper we used the following incrementally computed descriptors:

Area a. Area (i.e. number of pixels) of a region. The initialization function is a constant function $\psi(p) = 1$ and the combining operation $\oplus$ is an addition $(+)$.

Bounding box $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$. Top-right and bottom-left corner of the region. The initialization function of a pixel p with coordinates (x, y) is a quadruple $(x, y, x + 1, y + 1)$ and the combining operation $\oplus$ is $(\min, \min, \max, \max)$ where each operation is applied to its respective item in the quadruple. The width w and height h of the region is calculated as $x_{\max} - x_{\min}$ and $y_{\max} - y_{\min}$ respectively.

Perimeter p. The length of the boundary of the region (see Figure 4a). The initialization function $\psi(p)$ determines a change of the perimeter length by the pixel p at the threshold where it is added, i.e. $\psi(p) = 4 - 2|\{q : qAp \wedge \mathbf{C}(q) \leq \mathbf{C}(p)\}|$ and the combining operation $\oplus$ is an addition $(+)$. The complexity of $\psi(p)$ is $O(1)$, because each pixel has at most 4 neighbors.

Euler number η. Euler number (genus) is a topological feature of a binary image which is the difference between the number of connected components and the number of holes. A very efficient yet simple algorithm [18] calculates the Euler number by counting 2×2 pixel patterns called quads. Consider the following patterns of a binary image:

$$Q_1 = \left\{\begin{matrix}1 & 0 \\ 0 & 0\end{matrix}\right\}, \left\{\begin{matrix}0 & 0 \\ 0 & 0\end{matrix}\right\}, \left\{\begin{matrix}0 & 0 \\ 1 & 0\end{matrix}\right\}, \left\{\begin{matrix}0 & 0 \\ 1 & 1\end{matrix}\right\}, \left\{\begin{matrix}1 & 0 \\ 0 & 1\end{matrix}\right\}$$

$$Q_2 = \left\{\begin{matrix}0 & 1 \\ 1 & 1\end{matrix}\right\}, \left\{\begin{matrix}1 & 1 \\ 1 & 1\end{matrix}\right\}, \left\{\begin{matrix}1 & 1 \\ 1 & 0\end{matrix}\right\}, \left\{\begin{matrix}1 & 0 \\ 0 & 1\end{matrix}\right\}$$

$$Q_3 = \left\{\begin{matrix}0 & 1 \\ 1 & 0\end{matrix}\right\}, \left\{\begin{matrix}1 & 1 \\ 0 & 1\end{matrix}\right\}$$

Euler number is then calculated as

$$\eta = \frac{1}{4}\left(C_1 - C_2 + 2C_3\right))$$

where C_1, C_2 and C_3 denote number of quads Q_1, Q_2 and Q_3 respectively in the image.

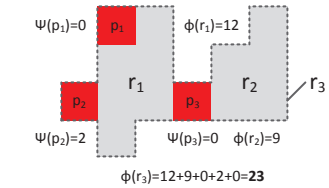
It follows that the algorithm can be exploited for incremental computation by simply counting the change in the number of quads in the image. The value of the initialization function $\psi(p)$ is determined by the change in the number of the quads Q_1, Q_2 and Q_3 by changing the value of the pixel p from 0 to 1 at given threshold $\mathbf{C}(p)$ (see Figure 4b), i.e. $\psi(p) = \frac{1}{4}\left(\Delta C_1 - \Delta C_2 + 2\Delta C_3\right))$. The complexity of $\psi(p)$ is $O(1)$, because each pixel is present in at most 4 quads. The combining operation $\oplus$ is an addition $(+)$.

Horizontal crossings $\mathbf{c}_i$. A vector (of length h) with number of transitions between pixels belonging ($p \in r$) and not belonging ($p \notin r$) to the region in given row i of the region r (see Figure 4c and 7). The value of the initialization function is given by the presence/absence of left and right neighboring pixels of the pixel p at the threshold $\mathbf{C}(p)$. The combining operation $\oplus$ is an element-wise addition $(+)$ which aligns the vectors so that the elements correspond to same rows. The computation complexity of $\psi(p)$ is constant (each pixel has at most 2 neighbors in the horizontal direction) and the element-wise addition has constant complexity as well assuming that a data structure with $O(1)$ random access and insertion at both ends (e.g. double-ended queue in a growing array) is used.

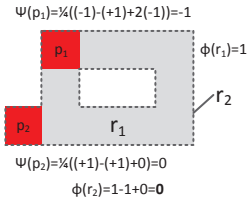
3.3. Sequential Classifier

In the proposed method, each channel is iterated separately (in the original and inverted projections) and subsequently ERs are detected. In order to reduce the high false positive rate and the high redundancy of the ER detector, only distinctive ERs which correspond to characters are selected by a sequential classifier. The classification is broken down into two stages for better computational efficiency (see Figure 2).

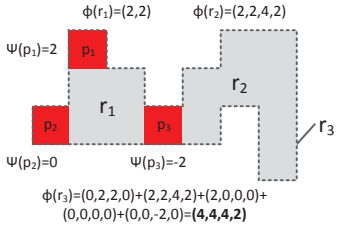
In the first stage, a threshold is increased step by step from 0 to 255, incrementally computable descriptors (see Section 3.2) are computed (in $O(1)$) for each ER r and the descriptors are used as features for a classifier which estimates the class-conditional probability $p(r|\text{character})$. The value of $p(r|\text{character})$ is tracked using the inclusion relation of ER across all thresholds (see Figure 6) and only the ERs which correspond to local maximum of the probability $p(r|\text{character})$ are selected (if the local maximum of the probability is above a global limit $p_{\min}$ and the difference between local maximum and local minimum is greater than $\Delta_{\min}$).



(a) perimeter p



(b) Euler number η



(c) Horizontal crossings $\mathbf{c}_i$

Figure 4. Incrementally computable descriptors. Regions already existing at threshold $\theta - 1$ marked grey, new pixels at threshold θ marked red, the resulting region at threshold θ outlined with a dashed line

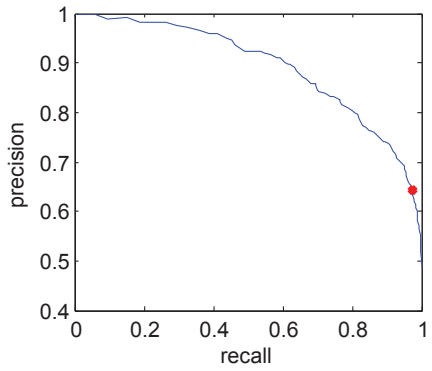
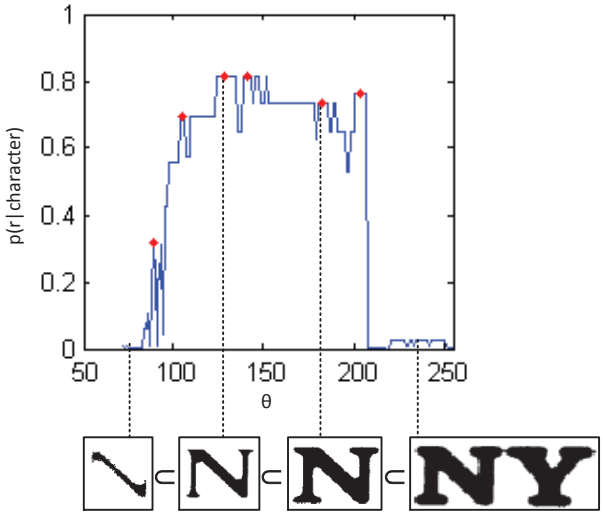


Figure 5. The ROC curve of the first stage of the sequential classifier obtained by cross-validation. The configuration used in the experiments marked red (recall 95.6%, precision 67.3)

In this paper, a Real AdaBoost [19] classifier with decision trees was used with the following features (calculated in $O(1)$ from incrementally computed descriptors): aspect ratio (w/h), compactness ($\sqrt{a}/p$), number of holes ($1 - \eta$) and a horizontal crossings feature ($\hat{c} =$ median $\{\mathbf{c}_{\frac{1}{6}w}, \mathbf{c}_{\frac{3}{6}w}, \mathbf{c}_{\frac{5}{6}w}\}$) which estimates number of character strokes in horizontal projection - see Figure 7. Only a fixed-size subset of $\mathbf{c}$ is sampled so that the computation has a constant complexity. The output of the classifier is calibrated to a probability function $p(r|\text{character})$ using Logistic Correction [16]. The parameters were set experimentally to $p_{\min} = 0.2$ and $\Delta_{\min} = 0.1$ to obtain a high value of recall (95.6%) (see Figure 5).

(a)



(b)

Figure 6. In the first stage of the sequential classification the probability $p(r|\text{character})$ of each ER is estimated using incrementally computable descriptors that exploit the inclusion relation of ERs. (a) A source image cut-out and the initial seed of the ER inclusion sequence (marked with a red cross). (b) The value of $p(r|\text{character})$ in the inclusion sequence, ERs passed to the second stage marked red

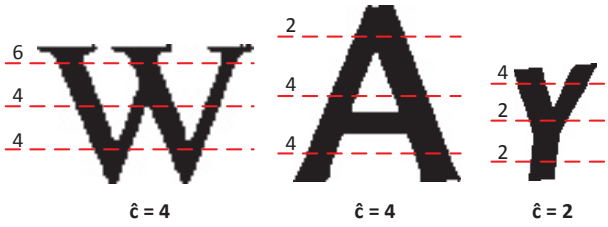


Figure 7. The horizontal crossings feature that is used in the 1st stage of ER classification

In the second stage, the ERs that passed the first stage are classified into character and non-character classes using more informative but also more computationally expensive features. In this paper, an SVM [4] classifier with the RBF kernel [13] was used. The classifier uses all the features calculated in the first stage and the following additional features:

- **Hole area ratio**. a_h/a where a_h denotes number of pixels of region holes. This feature is more informative

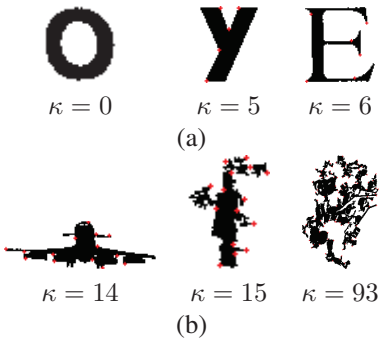


Figure 8. The number of boundary inflexion points κ. (a) Characters. (b) Non-textual content

than just the number of holes (used in the first stage) as small holes in a much larger region have lower significance than large holes in a region of comparable size.

- **Convex hull ratio**. a_c/a where a_c denotes the area of the convex hull of the region.

- **The number of outer boundary inflexion points** κ. The number of changes between concave and convex angle between pixels around the region border (see Figure 8). A character typically has only a limited number of inflexion points ($\kappa < 10$), whereas regions that correspond to non-textual content such as grass or pictograms have boundary with many spikes and thus more inflexion points.

Let us note that all features are scale-invariant, but not rotation-invariant which is why characters of different rotations had to be included in the training set.

3.4. Exhaustive Search

The detector was incorporated into a system described in Neumann and Matas [15], which uses efficiently pruned search to exhaustively search the space of all character sequences in real-time. It exploits higher-order properties of text such as word text lines and its robust grouping stage is able to compensate errors of the character detector. The system was chosen because it is able to handle multiple channels, multiple segmentations for each character (see Figure 6) and to combine detection results from multiple channels using the OCR stage. It also provides text recognition for characters segmented by the character detector. For more details see [15].

4. Experiments

The method was trained using approximately 900 examples of character ERs and 1400 examples of non-character ERs manually extracted the from the ICDAR 2003 training dataset [10] (sequential classifier training) and synthetically generated fonts (OCR stage training). The method was then evaluated with the same parameters on two independent datasets.

method	recall	precision	f
Kim's Method *	62.5	83.0	71.3
proposed method	**64.7**	**73.1**	**68.7**
Yi's Method [23]	58.1	67.2	62.3
TH-TextLoc System [8]	57.7	67.0	62.0
Neumann and Matas [15]	52.5	68.9	59.6

Table 2. Text localization results on the ICDAR 2011 dataset. Unpublished methods marked with a star

4.1. ICDAR 2011 Dataset

The ICDAR 2011 Robust Reading competition dataset [20] contains 1189 words and 6393 letters in 255 images. Using the ICDAR 2011 competition evaluation scheme [22], the method achieves the recall of 64.7%, precision of 73.1% and the f-measure of 68.7% in text localization (see Figure 9 for sample outputs).

The method achieves significantly better recall (65%) than the winner of ICDAR 2011 Robust Reading competition (62%), but the precision (73%) is worse than the winner (83%) and thus the resulting combined f-measure (69%) is worse than the ICDAR 2011 winner (71%), which had not been published. The proposed method however significantly outperforms the second best (published) method of Yi [23] in all three measures (see Table 2). Let us further note that the ICDAR 2011 competition was held in an open mode where authors supply only outputs of their methods on a previously published competition dataset.

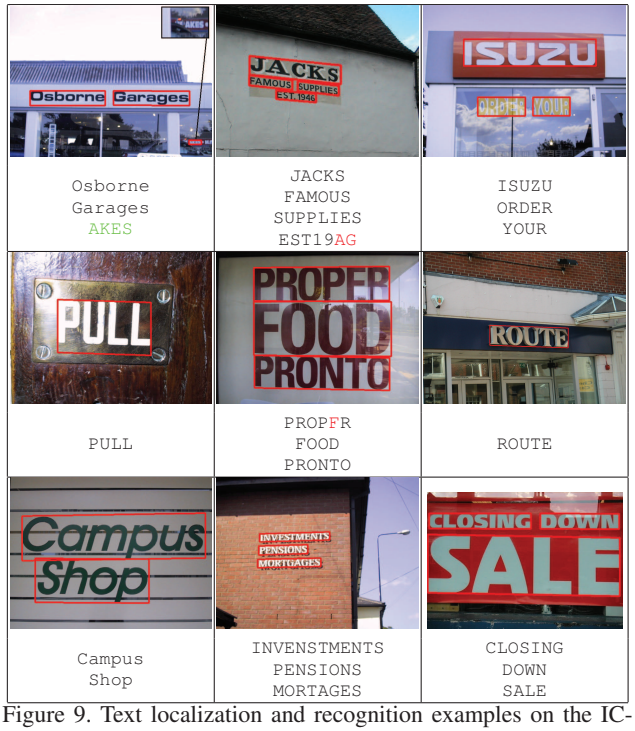
Word text recognition recall is 37.2%, precision 37.1% and f-measure 36.5% respectively; a word is considered correctly recognized if it is localized with recall at least 80% and all letters are recognized correctly (case-sensitive comparison) [10]. The average processing time (including text localization) is 1.8s per image on a standard PC.

The word recognition results cannot be compared to any existing method because end-to-end text localization and recognition was not part of the ICDAR 2011 Robust Reading competition and no other method had presented its text recognition results on the dataset.

4.2. Street View Text Dataset

The Street View Text (SVT) dataset [21] contains 647 words and 3796 letters in 249 images harvested from Google Street View. The dataset is more challenging because text is present in different orientations, the variety of fonts is bigger and the images are noisy. The format of the ground truth is different from the previous experiment as only some words are annotated (see Figure 11). Of the annotated words the proposed method achieved a recall of 32.9% using the same evaluation protocol as in the previous section (see Figure 12 for output examples).

The precision of the text localization (19.1%) cannot be

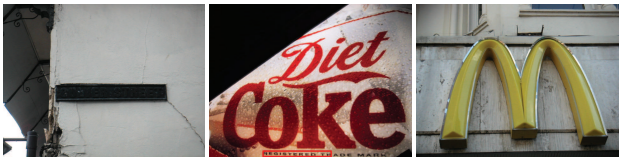


Osborne Garages AKES | JACKS FAMOUS SUPPLIES EST19AG | ISUZU ORDER YOUR

PULL | PROPFR FOOD PRONTO | ROUTE

Campus Shop | INVENSTMENTS PENSIONS MORTAGES | CLOSING DOWN SALE

Figure 9. Text localization and recognition examples on the ICDAR 2011 dataset. Notice the robustness against reflections and lines passing through the text (bottom-left). Incorrectly recognized letters marked red, text recognized by the proposed method but not present in the ground truth marked green



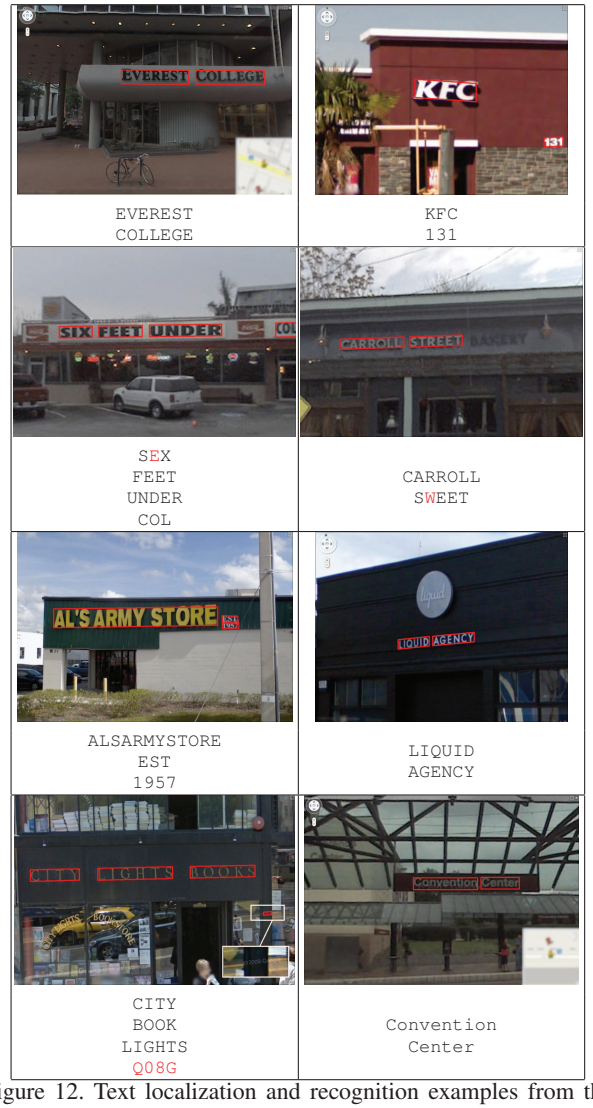
(a) (b) (c)

Figure 10. Problems of the proposed method. (a) Characters with no contrast. (b) Multiple characters joined together. (c) A single letter (the claim that the McDonald's logo is a letter "M" as defined by the annotation is questionable)

taken into account because of the incomplete annotations. It can be observed that many of the false detections are caused by watermark text embedded in each image (see Figure 1), which demonstrates robustness of the proposed method against noise and low contrast of characters.

The results can be compared only indirectly with the method of Wang et al. [21] which using a different evaluation protocol reports the f-measure of 41.0% (achieved for recall 29.0% and precision 67.0%) on the dataset. Moreover the task formulation of the method of Wang et al. differs significantly in that for each image it is given a lexicon of words that should be localized in the image (if present) whereas the proposed method has no prior knowledge about the content of the image and its output is not limited by a fixed lexicon.

Figure 11. Missing annotations in the SVT dataset (annotations marked green, output of the proposed method marked red).



EVEREST COLLEGE | KFC 131

SEX FEET UNDER COL | CARROLL SWEET

ALSARMYSTORE EST 1957 | LIQUID AGENCY

CITY BOOK LIGHTS Q08G | Convention Center

Figure 12. Text localization and recognition examples from the SVT dataset. Notice the high level of noise and the blur (zoomed-in PDF viewing highly recommended). Incorrectly recognized letters marked red.

5. Conclusions

An end-to-end real-time text localization and recognition method is presented in the paper. In the first stage of the classification, the probability of each ER being a character is estimated using novel features calculated with $O(1)$ complexity and only ERs with locally maximal probability are selected for the second stage, where the classification is improved using more computationally expensive features. It is demonstrated that including the novel gradient magnitude projection ERs cover 94.8% of characters. The average run time of the method on a 800×600 image is 0.3s on a standard PC.

The method was evaluated on two public datasets. On the ICDAR 2011 dataset the method achieves state-of-the-art text localization results amongst published methods (recall 64.7%, precision 73.1%, f-measure 68.7%) and we are the first to report results (recall 37.2%, precision 37.1%, f-measure 36.5%) for end-to-end text recognition on the ICDAR 2011 Robust Reading competition dataset.

On the more challenging Street View Text dataset the recall of the text localization (32.9%) can be only compared to the previously published method of Wang et al. [21] (29.0%), however direct comparison is not possible as the method of Wang et al. uses a different task formulation and a different evaluation protocol. Robustness of the proposed method against noise and low contrast of characters is demonstrated by "false positives" caused by detected watermark text in the dataset.

Acknowledgment

The authors were supported by The Czech Science Foundation Project GACR P103/12/G084.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[2] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. *CVPR*, 2:366–373, 2004.

[3] H. Cheng, X. Jiang, Y. Sun, and J. Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259 – 2281, 2001.

[4] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, March 2000.

[5] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR 2010*, pages 2963 –2970.

[6] L. Jung-Jin, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch. Adaboost for text detection in natural scene. In *ICDAR 2011*, pages 429–434, 2011.

[7] R. Lienhart and A. Wernicke. Localizing and segmenting text in images and videos. *Circuits and Systems for Video Technology*, 12(4):256 –268, 2002.

[8] H. Liu and X. Ding. Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes. In *ICDAR 2005*, pages 19 – 23 Vol. 1.

[9] S. M. Lucas. Text locating competition results. *Document Analysis and Recognition, International Conference on*, 0:80–85, 2005.

[10] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *ICDAR 2003*, page 682, 2003.

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22:761–767, 2004.

[12] J. Matas and K. Zimmermann. A new class of learnable detectors for categorisation. In *Image Analysis*, volume 3540 of *LNCS*, pages 541–550. 2005.

[13] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12:181–201, 2001.

[14] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *ACCV 2010*, volume IV of *LNCS 6495*, pages 2067–2078, November 2010.

[15] L. Neumann and J. Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *ICDAR 2011*, pages 687–691, 2011.

[16] A. Niculescu-Mizil and R. Caruana. Obtaining calibrated probabilities from boosting. In *In: Proc. 21st Conference on Uncertainty in Artificial Intelligence*, 2005.

[17] Y.-F. Pan, X. Hou, and C.-L. Liu. Text localization in natural scene images based on conditional random field. In *ICDAR 2009*, pages 6–10. IEEE Computer Society, 2009.

[18] W. K. Pratt. *Digital Image Processing: PIKS Inside*. John Wiley & Sons, Inc., New York, NY, USA, 3rd edition, 2001.

[19] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

[20] A. Shahab, F. Shafait, and A. Dengel. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In *ICDAR 2011*, pages 1491–1496, 2011.

[21] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *ICCV 2011*, 2011.

[22] C. Wolf and J.-M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *Int. J. Doc. Anal. Recognit.*, 8:280–296, August 2006.

[23] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. *Image Processing, IEEE Transactions on*, 20(9):2594 –2605, sept. 2011.

[24] J. Zhang and R. Kasturi. Character energy and link energy-based text extraction in scene images. In *ACCV 2010*, volume II of *LNCS 6495*, pages 832–844, November 2010.