

Efficient Scene Text Localization and Recognition with Local Character Refinement

Lukáš Neumann

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

Jiří Matas

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

Abstract—An unconstrained end-to-end text localization and recognition method is presented. The method detects initial text hypothesis in a single pass by an efficient region-based method and subsequently refines the text hypothesis using a more robust local text model, which deviates from the common assumption of region-based methods that all characters are detected as connected components.

Additionally, a novel feature based on character stroke area estimation is introduced. The feature is efficiently computed from a region distance map, it is invariant to scaling and rotation and allows to efficiently detect text regions regardless of what fragment of text they capture.

The method runs in real time and achieves state-of-the-art text localization and recognition results on the ICDAR 2013 Robust Reading dataset.

I. INTRODUCTION

Scene text localization and recognition, also known as text-in-the-wild or PhotoOCR, is an interesting problem with many application areas such as translation, assistance to the visually impaired or searching large image databases (e.g. Flickr or Google Images) by their textual content. But unlike traditional document OCR, none of the scene text recognition methods has yet achieved sufficient accuracy and speed for practical applications.

Text localization can be computationally very expensive because in an image of N pixels in general up to any of the 2^N subsets can correspond to text. Methods based on the sliding-window localize individual characters [1], [22] or whole words [5] by shifting a classification window of multiple sizes across the image, drawing inspiration from other object detection problems where this approach has been successfully applied [21]. Such methods are robust to noise and blur, since features aggregated over a larger area, but the crucial disadvantage is that the number of windows that needs to be classified grows rapidly when text with different scale, aspect, rotation and other distortions has to be found.

Methods based on connected components [3], [14]–[16], [19], [23] find individual characters as connected components of a certain local property (color, intensity, stroke-width, etc.), so that the complexity is unaffected by the text parameters as characters of all scales and orientations can be detected in one pass. Moreover, the connected component representation provides a segmentation which can be exploited in a standard OCR stage. The main drawback is the assumption that a character is a single connected component, which is brittle - a change in a single image pixel introduced by noise can cause an unproportional change in the connected component size,

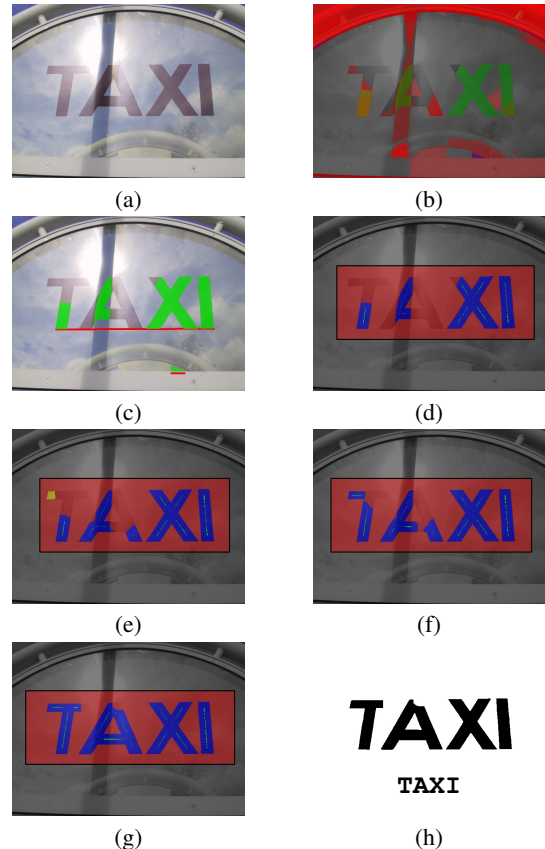


Fig. 1. The method pipeline. Source image (a). Initial MSER detection and classification (b) - character MSERs denoted green, multi-character MSERs blue and background MSERs denoted red. Text lines formation (c) - bottom line estimate in red. Local text refinement for the first text line - initialization (d), first iteration (e), second iteration (f), the last iteration (g), *definitive foreground* DF pixels in green, *probable foreground* PF pixels in blue, *background* B pixels in red, ignored pixels in yellow. Final segmentation and text recognition (h)

shape or other properties. The methods are also incapable of detecting characters which consist of several connected components or where text is present as characters joint together.

In the proposed method, we generalize the region-based approach by detecting arbitrary fragments and groups of characters alongside characters themselves in a single stage. As previously suggested [3], we benefit the observation that text consists of strokes and we propose an unified approach to effectively detect and further segment regions which are formed of strokes, regardless whether they represent a part of a character, a whole character or a group of characters

joint together, thus dropping the common assumption of a one to one correspondence between a character and its connected component representation.

In the initial stage, candidate regions are effectively detected as MSERs [10] with the “strokeness” property and grouped into initial text line hypotheses, where each text line hypothesis is then individually segmented or rejected using an iterative and more robust segmentation approach, which is capable of segmenting characters that cannot be obtained by thresholding (and therefore neither as MSERs). In order to estimate the “strokeness” of a region we introduce a novel feature based on *Stroke Support Pixels (SSPs)* which exploits the observation that one can draw any character by taking a brush with a diameter of the stroke width and drawing through certain points of the character (see Figure 3) - we refer to such points as *stroke support pixels (SSPs)*. The SSPs have the property that they are in the middle of a character stroke, which we refer to as the *stroke axis*, the distance to the region boundary is half of the stroke width, but unlike skeletons they do not necessary form a connected graph.

Since the area (i.e. the number of pixels) of an ideal stroke is the product of the stroke width and the length of the stroke, the “strokeness” can be estimated by the *stroke area ratio* feature ζ which compares the actual area of a region with the ideal stroke area calculated from the SSPs. The feature estimates the proportion of region pixels which are part of a character stroke and therefore it allows to efficiently differentiate between text regions (regardless of how many characters they represent - see Figure 5) and the background clutter. The feature is efficiently computed from a region distance map, it is invariant to scaling and rotation and it is more robust to noise than methods which aim to estimate a single stroke width value [3] as small pixel changes do not cause unproportional changes to the estimate. At last but not least, the SSPs are also exploited in the subsequent supervised segmentation stage to build a more accurate text color model, as by definition the SSPs are placed inside the character where the character color varies the least.

The rest of the paper is structured as follows: In Section II, an overview of previously published methods is presented, in the Section III the proposed method is introduced and in Section IV, the experimental evaluation is given. The paper is concluded in the Section V.

II. PREVIOUS WORK

Numerous methods which focus solely on text localization in real-world images have been published. The “sliding-window” based methods [9] use a window which is moved over the image and the presence of text is estimated on the basis of local image features. The majority of recently published methods for text localization however uses the connected component approach [3], [6], [11], [14], [15], [23]. The methods differ in their approach to individual character detection, which could be based on edge detection, character energy calculation or extremal region detection, but they all share the assumption that a single character is detected as a single connected component. The winning method in text localization of Yin et al. [25] at the latest ICDAR 2013 Robust Reading competition [7] also falls into this category as individual characters are detected as MSERs [10].

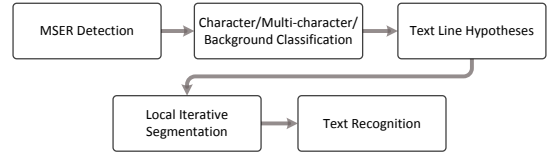


Fig. 2. Overview of the method. Initial text hypotheses efficiently generated by a MSER detector are further refined using a local text model

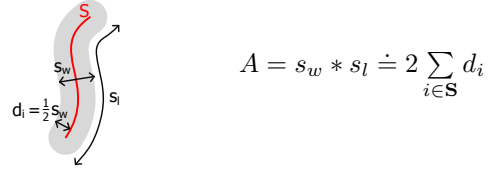


Fig. 3. Area A of an ideal stroke is a product of the stroke width s_w and the length of the stroke s_l . This is approximated by summing double the distances d_i of *Stroke Support Pixels (SSPs)* along the stroke axis S

Other methods focus only on text recognition, where the text is manually localized by a human annotator. The text is recognized on various levels, ranging from characters [4] to the whole word level [1], [8], [24]. The winning method [1] was able to correctly recognize 82.8% of the manually cropped-words in the latest ICDAR Robust Reading competition [7]. Although the methods for cropped-word recognition give an upper-bound on currently achievable text recognition performance, they in fact assume there exists a text localization method with a 100% accuracy, which currently is far from being true. Moreover, since the text was localized by a human, it is not clear that such text localization is even possible without the recognition, because the human annotator could have used the actual content of the text to create the annotation for localization.

For an exhaustive survey of text localization and recognition methods refer to the ICDAR Robust Reading competition results [7].

III. THE PROPOSED METHOD

A. Initial Candidates Detection

In the initial stage, candidate regions are detected as MSERs [10]. The MSER detector is often exploited in the literature [16], [25] to effectively detect individual characters, however this assumption may not always hold - there are many instances where individual characters cannot be detected as MSERs because only a fragment of a character is a MSER (see Figure 1b) or a single MSER corresponds to multiple characters or even whole words (see Figure 5, middle row).

In the proposed method, we significantly relax the assumption of individual characters being detected as MSERs (or even Extremal Regions [14]) by considering the MSER detector as an efficient first stage in order to generate initial text hypothesis, with no assumptions what level of text (i.e. part of characters, characters or words) individual MSERs represent. In other words, the proposed method assumes that at least a small portion of the text in the image triggers the MSER detector to generate an initial hypothesis, but it does require that all characters are detected as MSERs, as the individual characters are segmented at a later stage using a local text model.

In order to build initial text hypotheses, all MSERs in an image (detected in the intensity and hue channels) are first clas-

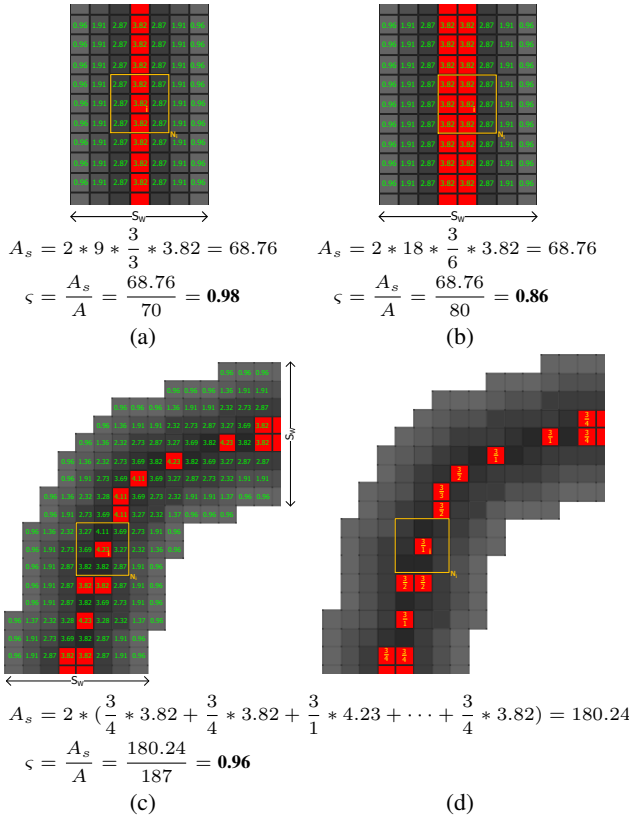


Fig. 4. Stroke area ratio ς calculation for a straight stroke of an odd (a) and even (b) width and for a curved stroke - distance map d_i (c) and Stroke Support Pixel weights w_i (d). Stroke Support Pixels (SSPs) denoted red

sified into 3 distinct classes: The *character* class represents a single character (or a significant character fragment), the *multi-characters* class represents an arbitrary group of characters joint together as a single component (e.g. a fragment of a word, a whole word or even several words) and the *background* class represents all non-textual content (e.g. background textures). The MSERs classified as *characters* and *multi-characters* are used to initialize a local text model (see Section III-C), whilst the MSERs classified as *background* are discarded.

For each region, the following features are calculated: stroke area ratio $\varsigma = \frac{A_s}{A}$, aspect ratio $\frac{w}{h}$, compactness $\frac{\sqrt{A}}{P}$, convex hull area ratio $\frac{A}{A_c}$ and holes area ratio $\frac{A_h}{A_c}$, where w and h denote width and height of the region's bounding box, A denotes the region area (i.e. number of pixels), P denotes the length of the perimeter, A_c denotes the convex hull area, A_h denotes the total area of region holes and A_s denotes the *character strokes area*.

In order to estimate the character strokes area, a distance transform map is calculated for the region binary mask and only pixels corresponding to local distance maxima are considered (see Figure 4) - we refer to these pixels as *Stroke Support Pixels (SSPs)*, because the pixels determine the position of a latent character stroke axis. In order to estimate the area of the character strokes \bar{A}_s , one could simply sum the distances associated with the SSPs

$$\bar{A}_s = 2 \sum_{i \in \mathbf{S}} d_i \quad (1)$$

where \mathbf{S} are the SSPs and d_i is the distance of the pixel i from the boundary.

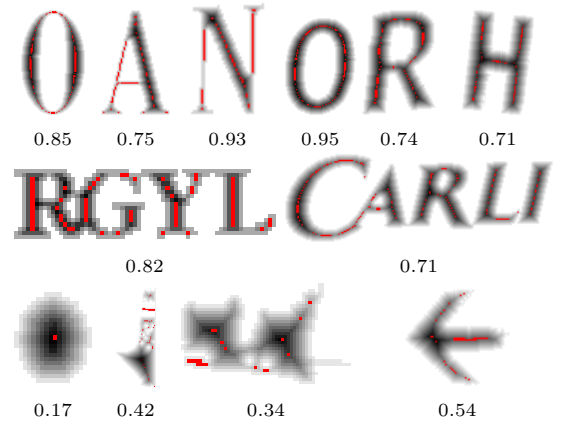


Fig. 5. Examples of stroke area ratio ς values for character (top row), multi-character (middle row) and background (bottom row) connected components. Distance map denoted by pixel intensity, *Stroke Support Pixels (SSPs)* denoted red

Such an estimate is correct for a straight stroke of an odd width, however it becomes inaccurate for strokes of an even width (because there are two support pixels for a unitary stroke length) or when the support pixels are not connected to each other as a result of stroke curvature, noise at the region boundary or changing stroke width (see Figure 4). We therefore propose to compensate the estimate by introducing a weight w_i for each SSP, which ensures normalization to a unitary stroke length by counting the number of pixels in a 3×3 neighborhood

$$A_s = 2 \sum_{i \in \mathbf{S}} w_i d_i \quad w_i = \frac{3}{|\mathcal{N}_i|} \quad (2)$$

where $|\mathcal{N}_i|$ denotes the number of SSPs within the 3×3 neighborhood of the pixel of i (including the pixel i itself). The numerator value is given by the observation that for a straight stroke, there are 3 support pixels in the 3×3 neighborhood (see Figure 4a).

To generate the training data, all MSERs from the ICDAR 2013 Training Set [7] dataset images were labeled using the ground truth segmentation masks - if the MSER overlaps sufficiently (more than 70% of pixels) with a ground truth character segmentation it is labeled as a *character*, if it overlaps with multiple ground truth character segmentations it is labeled as a *multi-character* and if it does not overlap with any segmentation it is labeled as *background*. MSERs which do not fall into any of the above categories were not used in the training. Using the aforementioned procedure, a dataset of 121,000 background MSERs, 14,000 character MSERs and 1,200 multi-character MSERs was obtained. A random subset of 20,000 samples was then used to train an SVM classifier [2] with a RBF [12] kernel using a one-against-all strategy, where each class was assigned a weight inversely proportional to its ratio in the training dataset in order to deal with the unbalanced number of samples for each class.

B. Text Line Hypotheses

Given the initial set of text hypothesis in the form of detected character and multi-character regions, the proposed method proceeds to build a local text model. The model is inferred for each *text line* individually, where we consider a *text line* as a sequence of characters which can be fitted by

a line and which has the same typographic and appearance properties.

The character and multi-character regions are first clustered into initial text line hypotheses using an efficient exhaustive search strategy adapted from [13], where each neighboring character triplet and each multi-character region is assigned a bottom line estimate (see Figure 1c), which serves as a distance measure for a standard agglomerative clustering approach. In order to enforce that one region is present only in one text line, initial text lines are simply grouped into clusters based on presence of identical regions (two text lines are a member of the same cluster if they have at least one region in common) and then in each cluster only the longest text line is kept; this can be viewed as a voting process, where in each cluster text lines vote for the text direction and the longest text line wins.

C. Local Iterative Segmentation

Each text line hypothesis is further refined using a local text model, individual for each text line. We formulate the problem of finding the local text model as a energy minimization task in a standard graph cut framework by adapting the iterative segmentation approach of GrabCut [17] by exploiting the SSPs for text color model training and by iteratively changing the processed image area based on the current segmentation.

Let us recall that in the graph cuts framework the objective is a minimization of a the Gibbs energy

$$E(\alpha, \theta, z) = U(\alpha, \theta, z) + V(\alpha, z) \quad (3)$$

where $U(\alpha, \theta, z)$ is the data term, $V(\alpha, z)$ is the smoothness term, α is the vector of labels for each pixel, θ represents the image color distributions for background and foreground and z is the image.

Following [17], the data term is a Gaussian Mixture Model (one GMM for foreground and one for background) and the smoothness term is based on the Euclidean distance in the RGB color space. Each pixel within the text line bounding-box is then labeled as *definitive foreground* DF , *probable foreground* PF or *background* B in the following iterative process (see Figure 1d-g):

- 1) Initialize all pixels belonging to a character or a multi-character region as PF , others as B
- 2) Calculate a new text line bounding-box by encapsulating all PF pixels and expand it by γ_h and γ_v pixels in the horizontal resp. vertical direction
- 3) Find SSPs amongst PF pixels and mark them as DF
- 4) Learn GMM parameters, using the DF pixels to train the foreground model and B pixels to train the background model
- 5) Create edges from the source to the DF and PF pixels, and from the B pixels to the sink
- 6) Estimate the segmentation by finding the minimal cut - mark pixels in the source subgraph as PF , pixels in the sink subgraph as B
- 7) Repeat from Step 2, until convergence

The value γ_h is set to the average region width in the text line and the γ_v is one third of the text line bounding-box height.

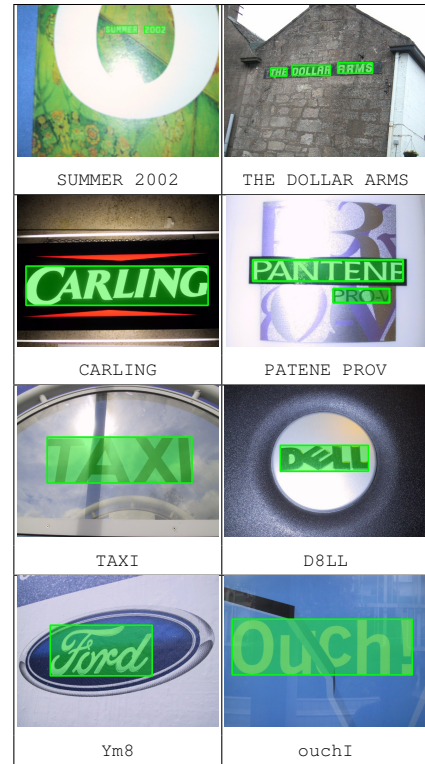


Fig. 6. Text localization and recognition examples on the ICDAR 2013 dataset

The final segmentation of the text line is obtained by taking the connected components of the PF pixels. If all pixels in the text line bounding-box converged to the same label (e.g. all are labeled as PF), the whole text line is discarded as it most likely represents a false positive. Pixels with the PF label which do not fit the bottom line estimate (see Figure 1e) or which are located at the boundary of the text line bounding-box are ignored in the Steps 2-4 as they typically represent interpunctuation or character fragments in neighboring text lines.

D. Text Recognition

Given the segmentations obtained in the previous stage, each connected component is assigned a Unicode label(s) by an OCR module, which is trained on synthetic data [16]. Following the standard approach of printed document OCR [20], the connected components with the aspect ratio above a predefined threshold are chopped to generate more region hypotheses in order to cater for joint characters. Each connected component with a label then represents a node in a direct acyclic graph, where the edges represent a “is-a-predecessor” relation. The final sequence of labels is then found as an optimal path in such a graph [15].

Because the graph is relatively small (when compared to [15], where there are several segmentations for each character), second order language model features were added in order to improve recognition accuracy without any significant impact on memory complexity.

The whole pipeline runs independently over multiple scales for each image and in the final stage the detected words are aggregated into a single output, while eliminating overlapping words (which typically represent the same word detected in

TABLE I. COMPARISON WITH MOST RECENT RESULTS ON THE ICDAR 2013 DATASET.

method	recall	precision	f	published
proposed method	72.4	81.8	77.1	
Yin et al. [25]	68.3	86.3	76.2	2014
TexStar (ICDAR'13 winner) [7]	66.4	88.5	75.9	2013
our previous method [15]	64.8	87.5	74.5	2013
Kim (ICDAR'11 winner) [18]	62.5	83.0	71.3	N/A

multiple scales) by keeping only the word whose corresponding path in the graph has the lowest cost.

IV. EXPERIMENTS

The proposed method was evaluated using the ICDAR 2013 Robust Reading competition dataset [7], which contains 1189 words and 6393 letters in 255 images.

Using the ICDAR 2013 competition evaluation scheme [7], the method achieves recall 72.4%, precision 81.8% and f-measure 77.1% in text localization (see Figure 6 for sample outputs). The method achieves significantly better recall than the winner of ICDAR 2013 Robust Reading competition (66%) and the method of Yin et al. [25] (68%). The precision is worse than some of the previous methods, but the overall f-measure represents the state-of-the-art performance - see Table I.

In end-to-end text recognition, the method correctly localized and recognized 543 words (45%), where a word is considered correctly recognized if all its characters match the ground truth (using case-sensitive comparison). On the other hand, the method “hallucinated” 79 words in total which do not have any overlap with the ground truth.

The main reasons for method failure are character-like objects near the text (e.g. pictographs, arrows, etc.) and low-contrast characters which are not picked up in the initial stage. The average run time on a standard 2.7GHz PC is 800ms per image.

V. CONCLUSION

An end-to-end real-time text localization and recognition method was presented. The method detects initial text hypothesis in a single pass by an efficient region-based method and subsequently refines the text hypothesis using a more robust local text model, which deviates from the common assumption of region-based methods that all characters are detected as connected components.

Additionally, a novel feature based on Stroke Support Pixels (SSPs) is introduced. The feature is based on an observation, that one can draw any character by taking a brush with a diameter of the stroke width and drawing through certain points of the character. The feature is efficiently computed from a region distance map, it is invariant to scaling and rotations and allows to efficiently detect text regions regardless of what portion of text they capture.

On the standard ICDAR 2013 dataset [7], the method achieves state-of-the-art results in text localization (f-measure 77.1%) and improves previously published results for end-to-end text recognition, with the average run time of 800ms per image.

ACKNOWLEDGMENT

The authors were supported by the Czech Science Foundation Project GACR P103/12/G084. Lukas would also like to acknowledge the support of the Google PhD Fellowship and the Google Research Award.

REFERENCES

- [1] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, “PhotoOCR: reading text in uncontrolled conditions.” ICCV, 2013.
- [2] N. Cristianini and J. Shawe-Taylor, *An introduction to Support Vector Machines*. Cambridge University Press, March 2000.
- [3] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *CVPR 2010*, pp. 2963–2970.
- [4] M. Iwamura, M. Tsukada, and K. Kise, “Automatic labeling for scene text database,” in *12th International Conference on Document Analysis and Recognition (ICDAR 2013)*, Aug. 2013, pp. 1397–1401.
- [5] L. Jung-Jin, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, “Adaboost for text detection in natural scene,” in *ICDAR 2011*, 2011, pp. 429–434.
- [6] L. Kang, Y. Li, and D. Doermann, “Orientation robust text line detection in natural images,” in *CVPR 2014*, 2014, pp. 4034–4041.
- [7] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. de las Heras *et al.*, “ICDAR 2013 robust reading competition,” in *ICDAR 2013*. IEEE, 2013, pp. 1484–1493.
- [8] C.-Y. Lee, A. Bhardwaj, W. Di, V. Jagadeesh, and R. Piramuthu, “Region-based discriminative feature pooling for scene text recognition,” in *CVPR 2014*. IEEE, 2014, pp. 4050–4057.
- [9] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, “Adaboost for text detection in natural scene,” in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, sept. 2011, pp. 429–434.
- [10] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, pp. 761–767, 2004.
- [11] A. Mishra, K. Alahari, and C. V. Jawahar, “Top-down and bottom-up cues for scene text recognition,” in *CVPR 2012*, june 2012, pp. 2687–2694.
- [12] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, “An introduction to kernel-based learning algorithms,” *IEEE Trans. on Neural Networks*, vol. 12, pp. 181–201, 2001.
- [13] L. Neumann and J. Matas, “Text localization in real-world images using efficiently pruned exhaustive search,” in *ICDAR 2011*, sept. 2011, pp. 687–691.
- [14] —, “Real-time scene text localization and recognition,” in *CVPR 2012*, 6 2012, pp. 3538–3545.
- [15] —, “On combining multiple segmentations in scene text recognition,” in *ICDAR 2013*. California, US: IEEE, August 2013, pp. 523–527.
- [16] —, “A method for text localization and recognition in real-world images,” in *ACCV 2010*, ser. LNCS 6495, vol. IV, November 2010, pp. 2067–2078.
- [17] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.
- [18] A. Shahab, F. Shafait, and A. Dengel, “ICDAR 2011 robust reading competition challenge 2: Reading text in scene images,” in *ICDAR 2011*, 2011, pp. 1491–1496.
- [19] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, “Scene text detection using graph model built upon maximally stable extremal regions,” *PR*, vol. 34, no. 2, pp. 107–116, 2013.
- [20] R. Smith, “An overview of the tesseract ocr engine.” in *ICDAR*, vol. 7, no. 1, 2007, pp. 629–633.
- [21] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [22] K. Wang, B. Babenko, and S. Belongie, “End-to-end scene text recognition,” in *ICCV 2011*, 2011.
- [23] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, “Detecting texts of arbitrary orientations in natural images,” in *CVPR 2012*, june 2012, pp. 1083–1090.
- [24] C. Yao, X. Bai, B. Shi, and W. Liu, “Strokelets: A learned multi-scale representation for scene text recognition,” in *CVPR 2014*, 2014, pp. 4042–4049.
- [25] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, “Robust text detection in natural scene images,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 5, pp. 970–983, May 2014.