# Text Localization based on Fast Feature Pyramids and Multi-resolution Maximally Stable Extremal Regions

Alessandro Zamberletti, Lucia Noce, and Ignazio Gallo

University of Insubria, Department of Theoretical and Applied Science

**Abstract.** Text localization from scene images is a challenging task that finds application in many areas. In this work, we propose a novel hybrid text localization approach that exploits Multi-resolution Maximally Stable Extremal Regions to discard false-positive detections from the text confidence maps generated by a Fast Feature Pyramid based sliding window classifier. The use of a multi-scale approach during both feature computation and connected component extraction allows our method to identify uncommon text elements that are usually not detected by competing algorithms, while the adoption of approximated features and appropriately filtered connected components assures a low overall computational complexity of the proposed system.

## 1 Introduction

Text localization from scene images has recently gained attention due to its potential application in various areas.

Using the categorization criteria of Pan *et al.* [1], algorithms for text localization can be classified as either region-based [1–4] or connected component CC-based [5–9]. Region-based methods exploit local features and sliding window classifiers to identify potential regions of text and build text confidence maps, while CC-based methods are based on the observation that text characters usually show uniform characteristics and therefore appear as stable connected components within the processed images.

Both of the previously mentioned approaches have disadvantages: region-based methods need to process the image in a multi-scale manner to obtain satisfying results, this usually causes those methods to be computationally expensive as they spend most of their processing time performing feature computation at the different scales. Moreover, sliding window classifiers for text localization are prone to false-positive errors as some local regions in scene images are virtually indistinguishable from text characters [10].

Most CC-based text localization methods [5–9] identify stable connected components using Maximally Stable Extremal Regions (MSER) [11]. Even though the basic assumption of CC-based algorithms is that text characters always appear as MSER, this does not always hold true, *e.g.* almost none of the published CC-based algorithms participating in ICDAR'13 [12] competition successfully


**Fig. 1.** Examples of uncommon and difficult text components successfully detected by the proposed method (images from ICDAR'03 and ICDAR'13).

detect blurred or uncommon (graffiti, company logos, *etc.*) text characters, as those elements either do not appear as stable connected components or are discarded due to their irregular geometric properties.

In this work, we pair Fast Feature Pyramids and Aggregated Channel Features [13] with Multi-resolution Maximally Stable Extremal Regions (MR-MSER) [14] to propose a hybrid algorithm for text localization that exploits the key ideas of region-based and CC-based methods but tries to overcome some of their previously mentioned limitations.

Without losing detection accuracy, in multi-scale approaches some image features (gradients, *etc.*) can be approximated from nearby scales within the same feature pyramid, instead of being explicitly computed at every level, to reduce by 2 orders of magnitude the time required to complete the feature computation process [13]. In our method, an approximated feature based classifier trained with natural, synthetic and semi-synthetic data, is used to efficiently build text confidence maps that are subsequently refined using MR-MSER.

Throughout our experiments we prove that MR-MSER excels at extracting entire words of text from scene images as single connected components, this also holds true for words composed by uncommon and difficult character fonts. We exploit this ability to discard false-positive text regions from the text confidence maps generated by the sliding window classifier.

In our system, most of the initially extracted MR-MSER are stacked and discarded; together with the use of approximated feature, this choice assures that the proposed method maintains an acceptable computational complexity even though it employs a multi-scale approach during both feature computation and connected component extraction.

As shown by the publicly available detection results for ICDAR'13 (some examples are provided in Fig. 1), despite its simplicity, the proposed approach succeeds where competing CC-based text localization methods usually fail, and achieves good results for ICDAR's Challenge 2 Task 1.[1]

[1] http://dag.cvc.uab.es/icdar2013competition/?ch=2&com=results
**Method:** iwrr2014.

## 2 Related Works

### 2.1 Region-based Text Localization

Among region-based text localization methods [1–4], the works that are closer to the proposed method are the ones of Pan *et al.* [1] and Wang *et al.* [4].

Pan *et al.* [1] build a text confidence map by processing images in a sliding window manner, using Waldboost and HOG features. The confidence map is used, together with other geometric features and a Multi-layer Perceptron, to compute the binary and the unary weights of a component neighbourhood graph built over a set of connected components extracted using Niblack's text binarization algorithm. CRF are used to filter out non-text components from the graph, while the remaining neighboring elements are clustered together into Minimum Spanning Trees to form text words. In our approach, we exploit a similar text confidence map to identify potential regions of text.

Wang *et al.* [4] perform end-to-end text recognition using Random Ferns and Pictorial Structures. The part of their work that is related to ours is the choice of using synthetic positive training data: roughly 1000 images are synthesized per character using 40 different fonts, adding Gaussian noise and applying random affine deformations (similarly to [2]). The classifier trained exclusively using synthetic positive data achieves the same F-score of a NN classifier trained with HOG features extracted from native data.

Another novel idea from [4] is the choice of extracting negative training samples from classes of Microsoft Research Cambridge Object Recognition Image Database (MSRC) [15]: classes like *buildings* and *countryside* resemble the background patterns of ICDAR's images and help in reducing the number of false-positive detections generated by sliding window classifiers.

### 2.2 CC-based Text Localization

Most CC-based text localization methods [5–7] either exploit MSER [11] to identify potential text components that are filtered and clustered together to form words, or use the Stroke Width Transform [10] algorithm to identify connected components having low intra-stroke variance [16].

Neumann *et al.* [17] proved that Extremal Regions extracted from multiple image channels cover almost 95% over ground-truth character annotations for ICDAR'11; however, to decrease the complexity of the system, only a sub-optimal subset of those channels is used in [17].

Other works focused on maximizing the effectiveness of MSER in terms of number of text elements successfully captured as stable components, *e.g.* Li *et al.* [6] showed that blurred and low quality characters become stable when extracting MSER from images incorporating gradient magnitude and intensity channels information.

Another technique for improving the coverage of MSER is the one of Forssén and Lowe [14]: a pyramid of images is built and MSER, called MR-MSER, are extracted at multiple scales (1 scale per octave). This multi-resolution approach

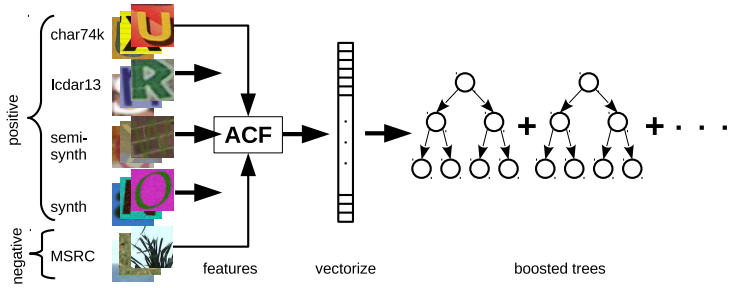



**Fig. 2.** Aggregated Channel Features (ACF) extracted from negative samples from MSRC and positive natural, synthetic and semi-synthetic samples from different datasets are used to train boosted depth-two decision trees.

causes some of the unstable regions in the original image to become stable at low scales in the pyramid, where the original image has lost most of its details as it has been sub-sampled and blurred multiple times with a Gaussian kernel.

In the proposed approach, we adopt the multiple image channel technique of [17] to extract words of text from scene images by computing multi-channel MR-MSER, appropriately filtering out useless regions extracted within the same pyramid's octave to keep an acceptable computational complexity.

### 2.3 Fast Feature Pyramids

Fast Feature Pyramids [13] revolutionized multi-scale sliding window approaches by showing that some image features (gradients, *etc.*) can be approximated from nearby scales within the same pyramid rather than being computed explicitly; since their introduction, Fast Feature Pyramids have been used in many works to build effective and efficient rigid object recognition detectors [18].

Based on the analysis of [19] on how to build the best classifier for rigid object recognition, we use a bootstrapped approximated feature based multi-scale linear classifier to perform text localization from scene images.

## 3 Proposed Model

The proposed approach is presented in this section: a binary classifier based on Fast Feature Pyramids and Aggregated Channel Features (Sec. 3.1) is trained using natural, synthetic and semi-synthetic data collected from multiple datasets or artificially generated (Sec. 3.2); predictions from the classifier are used to build a text confidence map in which potential regions of text are highlighted (Sec. 3.3); the text confidence map is used, together with MR-MSER (Sec. 3.4), to identify potential bounding boxes for lines of text in the processed image (Sec. 3.5).

An analysis of the computational complexity of the proposed approach and implementation details are provided in Sec. 3.6 and Table 1.


**Fig. 3.** MSER extracted at different levels of the pyramid capture different details: at low scales ($\leq 0.5$), characters are merged together and words are captured as single components, this also holds true for uncommon fonts (*e.g.* "Apocalypse Now"); in some instances, difficult characters that are not detected at the original scale are correctly identified as stable connected components at lower levels in the pyramid (*e.g.* "£99", graffiti).

### 3.1 Text Region Detector

The first step in our pipeline is to build a text confidence map by detecting potential regions of text using a multi-scale sliding window ACF detector [13].

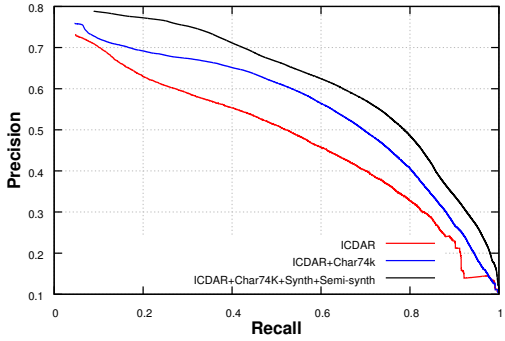
ACF uses Aggregated Channel Features, which are extracted by smoothing the processed image with a [1 2 1]/4 filter and then computing 10 different channels: normalized gradient magnitude, histogram of oriented gradients (6 orientations) and LUV. The channels are then condensed into 4×4 blocks and once again smoothed using the same approximated Gaussian kernel before being concatenated together to form single descriptors.

In our system, ACF is tuned to reach acceptable detection rates for text detection from scene images by setting the sliding window size to 32×32 pixels and the window stride to 16 pixels both in the horizontal and vertical directions. To deal with the large variation in size of text components in ICDAR datasets,

(a) ICDAR'13

(b) Synth

(c) Semi-synth




(d) PR graph

**Fig. 4.** Augmenting the positive training set with synthetic and semi-synthetic data increases the detection rate of the approximated feature based classifier.

we increase the size of the image pyramid by computing 1 octave above the canonical image scale. The final image pyramid goes from 2× the size of the original image $I$ to at most 32×32 pixel and has 8 scales per octave. For each octave, 7 scales out of 8 are approximated using $\lambda$ coefficients [13] inferred from 1000 random samples from the positive training set.

In our experiments, increasing the number of scales per octave or decreasing the number of approximated scales per octave did not affect the final results, on the other hand, decreasing the size of the image pyramid deeply affects the final detection rate, *e.g.* removing the highest octave while maintaining the same window size almost halves the accuracy of the classifier as tiny text components are not correctly detected. The same occurs when removing low pyramid levels or when improperly altering the size of the sliding window.

The ACF classifier is composed by 2048 quickly boosted [20] depth-two decision trees (3 stump classifiers per tree, as in Fig. 2).[2] Multiple rounds of bootstrapping are performed, at each round false-positive samples collected from the previous round are added to the negative training set, this shifts the decision boundary of the classifier and reduces false-positive detections in the subsequent round. Unlike [21], false-negative are not bootstrapped because text elements classified as background are usually identified using MR-MSER, as described in Sec. 3.5. Even when using 100000 samples and 3 rounds of bootstrapping, the classifier can be trained in less than 3 minutes on a Intel® Core i5 (see Table 1).

### 3.2 Training Data

Detection rates of linear classifiers are affected by both the quality/amount of training samples and the discriminative power of features extracted from those

[2] http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html


(a) source (b) *tcm* (c) *tex* (a) source (b) *tcm* (c) *tex*

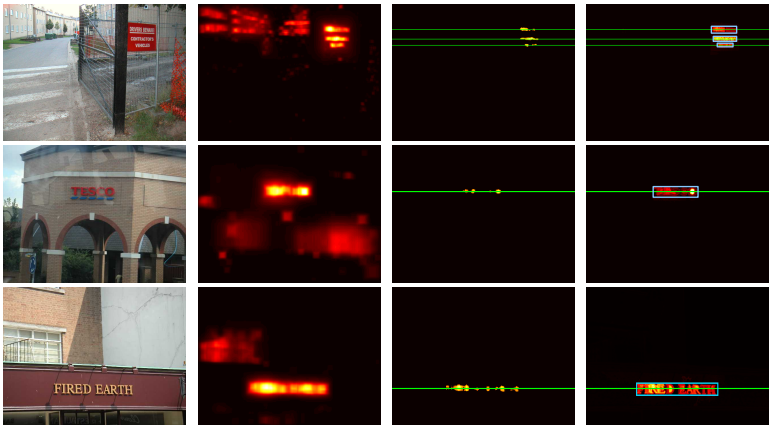
**Fig. 5.** True-positive regions discarded when thresholding the text confidence map *tcm* (b) are recovered in the textness map *tex* using MR-MSER (c).

samples. Considering that state-of-the-art results have been obtained in rigid object recognition by methods based on ICF and ACF [13, 18], we assume that good results may also be obtained in text detection using the same set of features when sufficient training data is collected. For this reason, we not only gather positive/negative samples from multiple datasets but we also generate additional semi-synthetic positive images by combining natural and synthetic images.

The process of extracting negative samples is straightforward: images not containing text are collected from some classes of MSRC database [15] (*benches*, *chairs*, *buildings*, *chimneys*, *kitchen utensils*, *miscellaneous*, *scenes*, *trees* and *windows*). In total, 1843 images containing only background components are gathered. For each image, a 4-level image pyramid (20%, 50%, 80% and 100% of the original image size) is built and 32×32 pixels patches are randomly extracted from all the pyramids, until a total of 50000 negative samples are gathered. Extracting negative examples at multiple scales reduces the number of false-positive detections generated at low octaves in the feature pyramid.

Gathering positive training samples is a challenging task, poor detection rates were obtained when training the ACF classifier using just the ≈5400 samples from ICDAR'13 [12] train set (see Fig. 4). To obtain acceptable performances, we augmented the set of positive training data with: ≈8000 images from the *GoodImg* class of Char74k English dataset [22], ≈6200 artificial images from the publicly available Synth dataset [4] (vertically cropped to remove neighbouring characters) and ≈30000 semi-synthetic samples obtained by combining natural background patches from MSRC images with synthetic fonts.

More in detail, semi-synthetic images are generated by placing random sized artificial characters in random positions over the images previously collected from MSRC to extract negative samples, random jitter (translation and rotation) is applied to increase the robustness of the classifier. Characters are cropped to their bounding boxes (leaving at most 5 pixels of random padding in every direction) and sub-sampled/up-sampled to 32×32 pixels. In order to keep an acceptable degree of contrast between the synthetic character and its surroundings, we compute the histogram of the patch on which the character is pasted and discard samples that are human unreadable (zero contrast between character and background).




(a) source img (b) *tcm* (c) thr. *tcm* (d) detection

**Fig. 6.** Text line formulation algorithm pipeline: the text confidence map (b) is thresholded (c) and words are identified using both the textness map *tex* and MR-MSER (d), final components are grouped together using Mean-shift.

Figure 4 shows how the positive sample sets we aggregate complement each others: samples from ICDAR train set and Char74k (Fig. 4a) contain uncommon and handmade characters that cannot be artificially generated; synthetic data from Synth (Fig. 4b) is useful to learn the shapes of artificial characters placed on plain backgrounds; semi-synthetic samples (Fig. 4c) are often placed on cluttered backgrounds and degraded due to sub-sampling/up-sampling, and thus represent a good connection point between synthetic and natural data.

### 3.3 Text Confidence Map

Let $\{s_0, \ldots, s_n\}$ be the scores assigned by the trained ACF classifier to each position of the sliding window in the image pyramid built for the processed image; a greedy Non-maximum Suppression (NMS) is performed to discard overlapping regions. In detail: (i) we discard regions having score lower than the average detection score $\mu(\{s_0, \ldots, s_n\})$; (ii) resize the remaining ones to half of their sizes to obtain a good separation between detected text regions, as in [4]; (iii) iterate over them by descending score and, if the region has not yet been suppressed, we suppress all the other non-suppressed regions having intersection-over-union $IoU > 0.5$ with the one currently selected; (iv) surviving regions are restored to their original sizes.

Using the suppressed regions we define a set of local text confidence maps $\{tc_0, \ldots, tc_j\}$, one for each level of the image pyramid. The final text confidence map *tcm* is obtained by stacking all the local confidence maps together $tcm = \frac{\sum_{i=1}^{j} tc_i}{n}$. Finally, *tcm* is normalized in $[0, 1]$ and thresholded at $t = 0.5$ to remove false-positive regions. True-positive regions discarded because of the threshold are recovered using MR-MSER, as explained in Sec. 3.5 (see Fig. 5).

### 3.4 Textness Map and MR-MSER

The text confidence map *tcm* is used, together with MR-MSER [14], to generate a *textness* map *tex* in which the value of each pixel denotes the probability it belongs to a text component in the original image $I$.

To extract MR-MSER, we compute 7 channels for $I$ (RGB, HSI and $\nabla$) and build an independent scale pyramid for each channel. MR-MSER are detected at each level of the pyramid, which has 1 octave per scale and a minimum size of 256×256 pixels; images in the pyramid are obtained by blurring and subsampling using a 6-tap Gaussian kernel with $\sigma = 1$. To reduce the final number of MR-MSER and discard the duplicate ones, at each level of each pyramid we retain only the larger MSER and filter out the smaller nested regions. This significantly decreases the final number of extracted MR-MSER: on average we discard more than 2000 regions from the ≈2500 initially identified.

Similarly to the text confidence map *tcm*, the *textness* map *tex* is built by iterating over the extracted MR-MSER and, for each of them, increasing the value of its pixels in the *tex* map by the average value of those pixels in *tcm*.

### 3.5 Text Line Formulation

The last step in a text localization framework consists in identifying the bounding boxes for words of text in the processed image; we formulate an algorithm that can be applied to different datasets without extensive tuning.

We propose a peak-based text grouping algorithm, in detail: (i) local maxima of the column-wise histogram of *tex* are identified, those peaks correspond to rows $\{r_0, \ldots, r_k\}$ of *tex* having maximum *textness* value compared to their neighbours; (ii) for each peak row $r_i$, connected components $\{cc_0, \ldots, cc_q\}$ intersecting $r_i$ in the text confidence map *tcm* are identified; (iii) each $cc_i$ is resized to the size of the minimum bounding box enclosing MR-MSER extracted from the image that have a pixel-wise $IoU > 0.2$ with $cc_i$; (iv) each resized $cc_i$ is assigned a score computed as the average intensity of its pixels in *tex*, and overlapping components are suppressed (as in Sec. 3.3); (v) neighbours connected components are merged into text lines using Mean-shift, components are clustered on the basis of their centroid positions. The pipeline is summarized in Fig. 6.

In phase (iii), we reshape regions labelled by the classifier as potential text areas according to the boundaries of MR-MSER extracted from the image. As it is possible to observe from Fig. 3 and Fig. 7a, MR-MSER extracted at low levels in the scale pyramid are often able to capture entire words (instead of single characters) as at those low levels most details of the original image are lost, and this causes characters to be merged together and words to be identified as single stable regions. Exploiting the word detection ability of MR-MSER to discard noise regions from the text confidence map without worrying about losing true-positive areas is the key idea of our method.

By grouping text components just on the basis of their centroid positions (ignoring scale, orientation, *etc.*), our algorithm can capture text in any possible orientation, even though ICDAR datasets do not contain examples of non-horizontal text components. The major drawback of ignoring orientation *etc.*

**Table 1.** Implementation details. Times refer to a 640×480 image and ≈500 MR-MSER processed on a desktop Intel® Core i5.

| Task | Time (s) | Implementation Type |
|---|---|---|
| Gathering p/n training data | 103.40 | Parallel |
| Training the classifier | 185.58 | Par. (Seq. feature computation) |
| Text confidence map | 0.29 | Parallel |
| Textness map | 0.45 | Parallel |
| Text line formulation | 0.01 | Parallel |

is that the proposed line formulation method frequently aggregates lines and phrases into single bounding boxes; such behaviour is penalized by some evaluation metrics (see Sec. 4.3), and additional processing may be required to split the detected bounding boxes into words before they are passed on to OCR engines.

### 3.6 Implementation Details

Timings information for the proposed approach are given in Table 1: gathering positive/negative samples and training the classifier for ICDAR'13 dataset require less than 5 minutes on a desktop Intel® Core i5 with 12 GB RAM.

On average, a 640×480 image can be fully processed in ≈0.75 seconds. The computational complexity of the method can be further reduced by decreasing the number of channels from which MR-MSER are extracted, at the cost of sacrificing the accuracy of the whole system.

Using the classifier configuration of Sec. 3.1 and the training data from Sec. 3.2, RAM consumption during training is ≈6 GB. On average, ≈500 MB of RAM are sufficient to build the *textness* map for a 640×480 image.

## 4 Experiments

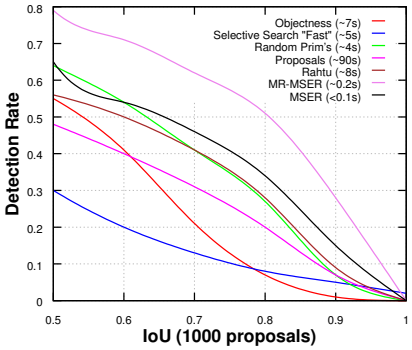
In this section, we provide an experimental evaluation of the components described in Sec. 3: the detection rate of the ACF text region detector introduced in Sec. 3.1 is evaluated in Sec. 4.1; MR-MSER are compared with MSER at detecting entire words of text in Sec. 4.2; text localization results achieved by the proposed approach for ICDAR'03 and ICDAR'13 datasets are presented in Sec. 4.3 and compared with competing published algorithms.

### 4.1 Classifier and Training Data

Figure 4d shows the PR curves for multiple ACF classifiers trained using the same parameter configuration but different training samples. PR curves are computed as in [2]: the text confidence map *tcm* is thresholded multiple times to yield binary decisions at each pixel and compared pixel-wise with ground-truth annotations from ICDAR'13 test set.

| Image Channels | MR-MSER | | MSER | |
|---|---|---|---|---|
| | chars | words | chars | words |
| $\nabla$ | 0.56 | 0.69 | 0.52 | 0.56 |
| RGB | 0.63 | 0.40 | 0.56 | 0.25 |
| HSI | 0.62 | 0.51 | 0.56 | 0.36 |
| HSI $\cup$ $\nabla$ | 0.71 | 0.77 | 0.67 | 0.65 |
| RGB $\cup$ $\nabla$ | 0.72 | 0.73 | 0.68 | 0.61 |
| HSI $\cup$ RGB | 0.70 | 0.56 | 0.64 | 0.41 |
| HSI $\cup$ RGB $\cup$ $\nabla$ | 0.75 | 0.78 | 0.71 | 0.66 |

(a) MR-MSER vs. MSER




(b) Word detection accuracy

**Fig. 7.** Evaluation of MR-MSER for word detection: (a) MR-MSER are compared with MSER at detecting single characters and entire words, while varying the image channels from which they are extracted, as in [17]; (b) word detection accuracy evaluation and timings information for MR-MSER, MSER and object proposal methods on ICDAR'03, while varying the intersection-over-union $IoU$ coverage tolerance.

This experiment shows how training data affects the performance of the ACF classifier: unsatisfying detection rates are obtained when training using just the samples from ICDAR'13 train set; significantly better results are obtained when combining ICDAR'13 train data with samples from Char74k; acceptable detection accuracies are achieved when augmenting the positive training set with synthetic and semi-synthetic data collected as described in Sec. 3.2.

In every experiment the training set has been kept balanced, meaning that the number of negative samples has always been equal to the number of positive samples. AUC of the PR curve for the classifier trained with natural, synthetic and semi-synthetic data is higher than the one of [2], proving the effectiveness of approximated feature based sliding window classifiers for text localization.

### 4.2 Word Detection via MR-MSER

The proposed method relies on the ability of MR-MSER to identify entire words of text from scene images (see Sec. 3.4 and Sec. 3.5).

In Fig. 7a, MR-MSER and MSER are compared at the task of identifying single characters (*chars*) and entire words (*words*) on ICDAR'03, while varying the image channels from which MR-MSER and MSER are extracted. In our experiment, a character/word is considered identified if there exists at least one MR-MSER/MSER, from the ones extracted, whose bounding box has an intersection-over-union $IoU > 0.5$ with the ground-truth annotation of that character/word.

Even though in this experiment we compared filtered MR-MSER (smaller nested regions are discarded as described in Sec. 3.4) with unfiltered MSER (all

the regions are retained for evaluation), MR-MSER outperform MSER both at detecting single characters and entire words for all the evaluated combinations of image channels. On average, the extraction of MR-MSER requires 0.2 sec using a parallel implementation (multiple scales within the same octave are extracted at the same time), while the computation of MSER requires 0.1 sec per image.

In order to measure how accurately MR-MSER detect entire words of text at low blurred scales in the image pyramid, in Fig. 7b they are evaluated on ICDAR'03 while varying the $IoU$ coverage tolerance. Since text characters and words satisfy some of the conditions analysed in [23], object proposal methods have also been added to the comparison to see whether they constitute a valid alternative to MR-MSER or MSER at detecting words of text in scene images. Results are measured as in [24]: for each algorithm, at most 1000 bounding boxes per image are selected from the ones initially extracted; Detection Rate (DR, $y$-axis) is the percentage of ground-truth words *covered* by those bounding boxes. A ground-truth word is *covered* if there exists at least one bounding box, among the 1000 selected, that has an $IoU > x$ with the ground-truth bounding box of that word. The value of $x$ varies on the $x$-axis, by increasing $x$ we require the identified bounding boxes to match more precisely the ground-truth data in order for a word to be considered *covered*.

The comparison is carried out as follows: (i) Objectness: among the ≈1850 ranked proposals generated per image, the top 1000 are selected for evaluation. MS, CC and SS cues are learnt from 50 images from ICDAR'03 training set; (ii) Selective Search: evaluated in its *fast* variant, 1000 windows are uniformly sampled from the ones initially extracted; (iii) Prims: a grid search is performed in $[0, 5]$ for color similarity, common border ratio and size, the parameters providing the best results for 1000 unique windows and $IoU > 0.5$ are used for evaluation; (iv) Proposals: evaluation is performed considering the bounding boxes surrounding the identified ranked segmentations proposals, top 1000 windows are selected from the ones initially extracted; (v) MR-MSER: extracted as described in Sec. 3.4, the bounding boxes surrounding each MR-MSER are considered for evaluation, on average, no more than 500 windows per image are generated; (vi) MSER [11]: extracted from RGB, HSI and $\nabla$ channels, the bounding boxes surrounding 1000 unique MSER are uniformly sampled from the initial set. For references to the evaluated object proposals algorithms see [24].

MR-MSER prove their effectiveness as robust word detector from scene images by achieving higher detection accuracies throughout all the tolerance values.

### 4.3 Text Localization Results

In Tables 2a and 2b, the proposed text localization approach is evaluated on ICDAR'03 and ICDAR'13 datasets.

ICDAR 2003 [28] contains a total of 509 images: 258 for training and the remaining 251 for testing. The classifier is trained using 45000 positive samples from ICDAR'03 train set, Char74k, Synth and Semi-synth and 45000 negative samples from MSRC. Precision, Recall and F-score are computed by looking for the best match between each detected bounding boxes and each ground-truth

**Table 2.** Text localization results for ICDAR's Challenge 2 Task 1.

(a) ICDAR'03. Evaluation metric: [28]

| Method | Precision | Recall | F-score |
|---|---|---|---|
| Li [6] | 0.79 | 0.64 | 0.71 |
| Kim [5] | 0.78 | 0.65 | 0.71 |
| **Proposed** | **0.71** | **0.74** | **0.70** |
| TD-Mixture [16] | 0.69 | 0.66 | 0.67 |
| Yi [25] | 0.73 | 0.67 | 0.66 |
| Epshtein [10] | 0.73 | 0.60 | 0.66 |
| Li | 0.62 | 0.65 | 0.63 |
| Chen | 0.60 | 0.60 | 0.58 |
| Neumann [7] | 0.59 | 0.55 | 0.57 |
| Zhang | 0.67 | 0.46 | 0.55 |

(b) ICDAR'13. Evaluation metric: [12]

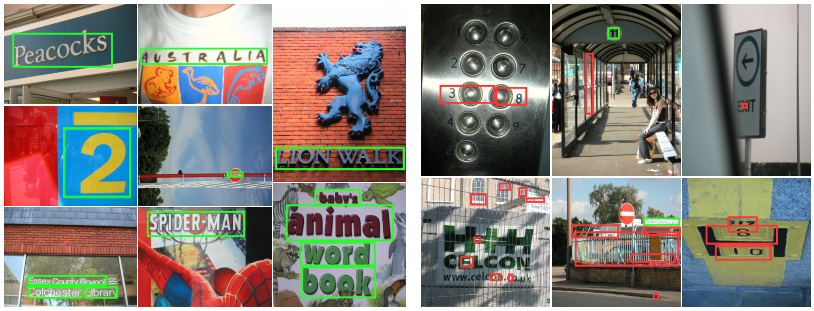
| Method | Precision | Recall | F-score |
|---|---|---|---|
| **Proposed** | **0.86** | **0.70** | **0.77** |
| Yin [9] | 0.88 | 0.66 | 0.76 |
| Neumann [26] | 0.88 | 0.65 | 0.74 |
| Bai [27] | 0.79 | 0.68 | 0.73 |
| Shi [8] | 0.85 | 0.63 | 0.72 |
| Shijian | 0.75 | 0.69 | 0.72 |
| Yang | 0.70 | 0.65 | 0.67 |
| Fabrizio | 0.74 | 0.53 | 0.62 |
| Baseline | 0.61 | 0.35 | 0.44 |
| Inkam | 0.31 | 0.35 | 0.33 |

annotation [5]. This evaluation metric penalizes approaches that detect text at line level, as only *one-to-one* matches are taken into account. Since our method often captures entire phrases as single components, it generates numerous *many-to-one* detections and therefore performs slightly worse than [5, 6].

ICDAR 2013 [12] contains a total of 462 images: 229 for training and 233 for testing. The classifier is trained using 50000 positive samples from ICDAR'13 train set, Char74k, Synth and Semi-synth and 50000 negative samples from MSRC. Unlike ICDAR'03, results are measured using a new evaluation metric [12], which takes into account *one-to-one*, *one-to-many* and *many-to-one* matches between ground-truth annotations and detected bounding boxes. The competition protocol penalizes methods that perform text localization at character level (*one-to-many*) but does not inflict any penalty to methods that provide text detection at line level (*many-to-one*). F-score of the proposed method is higher than competing approaches both when measured using ICDAR's evaluation metric or DetEval [29]. Complete results are available on ICDAR's web page. For references to all the evaluated algorithms see [12, 28].

Using classic MSER in place of MR-MSER, F-score of the proposed method decreases by roughly 10% on both datasets, as expected from the analysis of Sec. 4.2, where multi-channel MR-MSER covers 78% of ICDAR's ground-truth words while MSER provides a coverage of 66%.

Negative detection results are provided in Fig. 8, the proposed method fails when MSER extracted at multiple scales do not capture text components or when the text confidence map is noisy and text components are lost due to threshold (*e.g.* "HHH CELCON"). It is in fact possible to obtain different values of Precision/Recall by shifting the threshold value used during the text confidence map building phase described in Sec. 3.3: lower threshold values increase the Recall of the algorithm and decrease its Precision, while higher values discard more components from the text confidence map and therefore decrease the Recall of the whole system while increasing its overall Precision.




(a) Positive (b) Negative

**Fig. 8.** Examples of positive and negative text localization results for ICDAR'13.

## 5 Conclusion

A novel method for text localization from scene images has been proposed, it exploits both the latest advancements in rigid object recognition and MR-MSER to obtain good results for text localization from scene images. In the proposed solution, stable connected components are not discarded on the basis of their geometric properties; this assures that uncommon text fonts that are typically filtered out as noise elements by competing approaches are correctly retained and identified. Thanks to the use of approximated multi-resolution features and appropriately filtered connected components extracted in a multi-scale multi-channel manner, the proposed system is computationally efficient to train and test. This enables its application to numerous problems in which execution and training times are critical factors.

## References

1. Pan, Y.F., Hou, X., Liu, C.L.: Text localization in natural scene images based on conditional random field. In: Proc. ICDAR. (2009)
2. Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D.J., Ng, A.Y.: Text detection and character recognition in scene images with unsupervised feature learning. In: Proc. ICDAR. (2011)
3. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: Proc. BVMC. (2012)
4. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: Proc. ICCV. (2011)
5. Koo, H.I., Kim, D.H.: Scene text detection via connected component clustering and non-text filtering. IEEE Trans. on IP **22** (2013) 2296–2305
6. Li, Y., Jia, W., Shen, C., Hengel, A.: Characterness: An indicator of text in the wild. IEEE Trans. on IP **23** (2014) 1666–1677
7. Neumann, L., Matas, J.: A method for text localization and recognition in real-world images. In: Proc. ACCV. (2011)

8. Shi, C., Wang, C., Xiao, B., Zhang, Y., Gao, S.: Scene text detection using graph model built upon maximally stable extremal regions. Pattern Recogn. Lett. **34** (2013) 107–116
9. Yin, X.C., Yin, X., Huang, K.: Robust text detection in natural scene images. IEEE Trans. on PAMI **36** (2013) 970–983
10. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Proc. CVPR. (2010)
11. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proc. BMVC. (2002)
12. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L., Mestre, S., Mas, J., Mota, D., Almaz, J., Heras, L.: ICDAR 2013 robust reading competition. In: Proc. ICDAR. (2013)
13. Dollár, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Trans. on PAMI **36** (2014) 1532–1545
14. Forssén, P.E., Lowe, D.G.: Shape descriptors for maximally stable extremal regions. In: Proc. ICCV. (2007)
15. Crimisi, A.: Microsoft Research Cambridge Object Recognition Image Database. (2004)
16. Yao, C., Bai, X., Liu, W., Ma, Y.: Detecting texts of arbitrary orientations in natural images. In: Proc. CVPR. (2010)
17. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Proc. CVPR. (2012)
18. Mathias, M., Timofte, R., Benenson, R., Gool, L.V.: Traffic sign recognition: How far are we from the solution? In: Proc. IJCNN. (2013)
19. Benenson, R., Mathias, M., Tuytelaars, T., Gool, L.V.: Seeking the strongest rigid detector. In: Proc. CVPR. (2013)
20. Appeal, R., Fuchs, T., Dollár, P., Perona, P.: Quickly boosting decision trees pruning underachieving features early. In: Proc. ICML. (2013)
21. Villamizar, M., Andrade-Cetto, J., Sanfeliu, A., Moreno-Noguer, F.: Bootstrapping boosted random ferns for discriminative and efficient object classification. Pattern Recogn. **45** (2012) 3141–3153
22. de Campos, T.E., Babu, B.R., Varma, M.: Character recognition in natural images. In: Proc. VISAPP. (2009)
23. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: Proc. CVPR. (2010)
24. Manen, S., Guillaumin, M., Gool, L.V.: Prime object proposals with randomized prims algorithm. In: Proc. ICCV. (2013)
25. Yi, C., Tian, Y.: Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification. IEEE Trans. on IP **21** (2012) 4256–4268
26. Neumann, L., Matas, J.: On combining multiple segmentations in scene text recognition. In: Proc. ICDAR. (2013)
27. Bai, B., Yin, F., Liu, C.L.: Scene text localization using gradient local correlation. In: Proc. ICDAR. (2013)
28. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: ICDAR 2003 robust reading competition. In: Proc. ICDAR. (2003)
29. Wolf, C., Jolion, J.M.: Object count/area graphs for the evaluation of object detection and segmentation algorithms. IJDAR **8** (2006) 280–296