University of Insubria
Department of Theoretical and Applied Science (DiSTA)
PhD Thesis in Computer Science
XXVIII Cycle of Study

Text Localization and Recognition in Natural Scene Images

Candidate:
ALESSANDRO ZAMBERLETTI

Thesis Advisor:
Prof. IGNAZIO GALLO

December, 2015

To my parents, Angela e Riccardo

Contents

Acknowledgements

First and foremost, my greatest thanks goes to Prof. Ignazio Gallo (Università degli Studi dell'Insubria) and my colleague Lucia Noce for helping me during my research activities and making every day of my PhD course enjoying and relaxing.

I would also like to express my gratitude to Prof. Pierluigi Gallo (Università degli Studi di Palermo) and Prof. Gabriele Piccoli (Louisiana State University and Università degli studi di Pavia) for investing their time in reviewing this thesis and for providing me with invaluable feedback.

Special thanks also goes to Nicola Lamberti and Andrea Broglia, and to all the people that have been working in 7pixel Varese offices throughout these years. Thanks for letting me carry my research activities in your offices, for providing me with all the computational power needed to successfully develop my applications, and most importantly, for all the funny moments we have spent together.

Last but not least, a huge thanks to my family for having always been there for me every single day of my life, to my girlfriend, and to all the people that helped me during the difficult situations I faced in these last months.

Alessandro Zamberletti
Varese, 10 December 2015

1 Introduction

Text localization and recognition (also known as *text spotting*) in natural scene images is an interesting task that finds many practical applications.

For instance, an algorithm for text localization and recognition may be used in helping visually impaired subjects during navigation in unknown enviroments, building autonomous driving systems that automatically avoid collisions with pedestrians or automatically identify speed limits or other driving rules and warn the driver about possible driving infractions that are being committed, and to ease or solve some tedious and repetitive data entry tasks that are still carried out manually by humans.

While Optical Character Recognition (OCR) from scanned digital documents may be considered a solved problem since state-of-the-art methods reach roughly 99% text reading accuracy for that class of images, the same cannot be said for natural scene images. In fact, this latest class of images cointains plenty of difficult situations that algorithms for text localization and reading need to deal with in order to reach acceptable text reading accuracies, and decent text localization detection rates.

For example, unlike algorithms for OCR from digital scanned documents, in which a simple scaling and/or rotation of the images may be enough to correctly recognize the text, in natural scene images the algorithm should also cope with plenty of more difficult situations commonly found in real world images acquired with mobile phones or webcams, such as: perspective projection distortions, light variations and reflections, variable and unknown distances from the camera lens, total or partial occlusions, uncommon fonts like company logos or uncommon text characters like those found in arabic

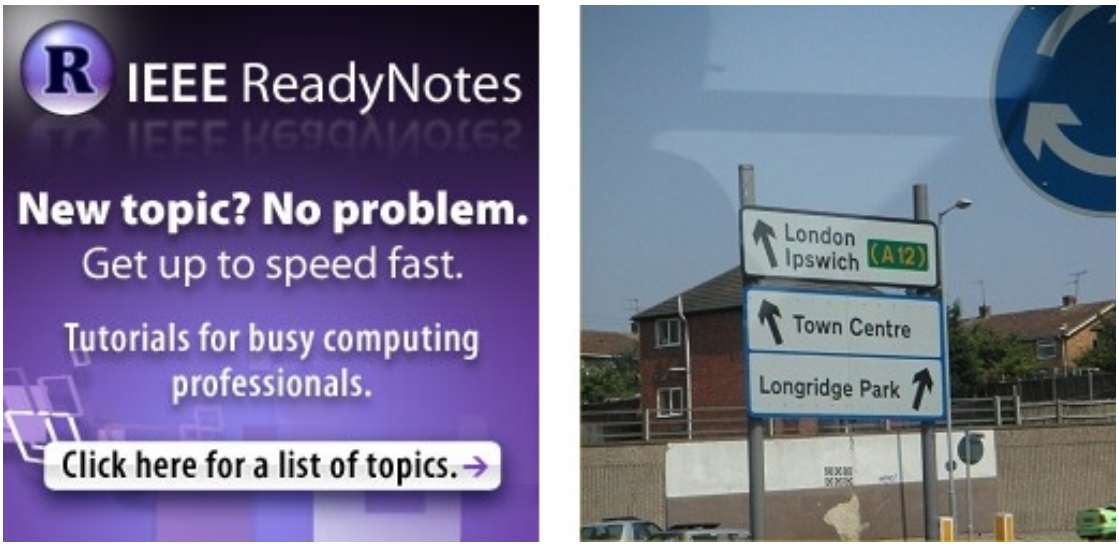


Figure 1.1: There is a huge difference between text characters found in scanned born-digital images (left image, from ICDAR 2015 Challenge 1 Task 4) and natural scene images (right image, from ICDAR 2015 Challenge 2 Task 4). Scaling the left image may lead to good text localization/recognition results; while the right one also requires warping to sucessfully detect text words.

texts, to name a few.

Many of these difficult conditions are found in images from standard text localization and recognition datasets and failing to deal with or ignoring any of these issues will cause the accuracies or detection rates of the text localization/recognition algorithm that is being developed to significantly decrease.

Dealing with all those difficult situations is a trivial task, and different techniques for text localization and reading in natural scene images have been proposed in literature over the past decade to try to reach human performances in real applications. Most of the interesting and relevant related works recently presented in literature are presented and discussed in Chap. 2; starting from traditional, and usually computationally expensive, region-based methods which use sliding-window classifiers to build the so-called *text confidence maps*, all the way up to more sophisticated and efficient approaches based on stable connected-component analysis and/or deep architectures.

Most of the works cited in this thesis were presented after 2010, and the majority of them deal with either text localization, text reading, or end-to-end text localization and reading in natural scene images. This is a strong indication of the amount of time, resources and energies that Computer Vision researchers are investing on trying to increase state-of-the-art results for this task. A similar trend can be seen when looking

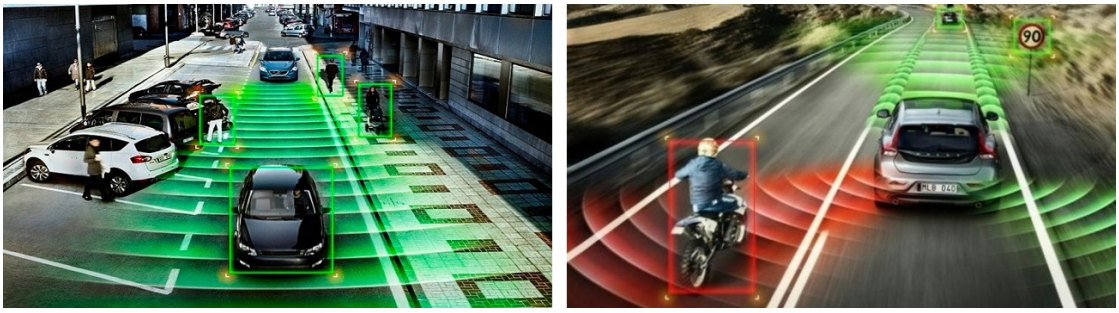


Figure 1.2: Autonomous driving systems (autonomous vehicles) are one of the most interesting and powerful practical application of end-to-end text localization and reading algorithms in natural scene images/videos. Automatically recognizing and reading traffic signs helps the driver to avoid unauthorized or dangerous driving maneuvers which may injury the driver and other people, or cause driving tickets.

at the amount of works on pedestrian detection published over the last five years. [1]

The main reason behind the increasing popularity of these research areas can be probably found in the recent autonomous driving systems that are being developed by big companies (Google Cars, *etc.*): increasing the affidability of text/pedestrian detection algorithms will reduce the usage of expensive sensors and the risk of collisions with pedestrians and other vehicles (Fig. 1.2).

The two main works on text localization and end-to-end text localization/recognition in natural scene images developed during my three years of PhD course are presented in this thesis. The construction of the text localization and the text spotting algorithms of Chap. 3 and 4 were done in collaboration with my colleague Lucia Noce and advised by Prof. Ignazio Gallo. Thus, when describing the algorithms and discussing about them, I generally refer to and talk about those works using first-person plural to reflect the joint nature of this work.

In our work of Chap. 3 we try to address most of the previously mentioned natural scene image problems (especially the localization of uncommon fonts and writings) by proposing a hybrid system which exploits both the key ideas of region-based methods and stable connected components extracted from the processed images. During the development of this method, our focus has always been on maintaining an acceptable computational complexity and a high reproducibility of the achieved text localization results. To this end, we exploited the latest advancements in Computer Vision (especially in pedestrian detection), using novel techniques and algorithms like Approximated Features and Aggregated Channel Feature Classifiers [2] which provide a good compromise between detection rates and computational complexity. As shown on the official web-

[1] State-of-the-art detection rates on "Caltech Pedestrians USA" dataset skyrocketed during ICCV 2015 [1] - http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

site of the International Conference of Document Analysis and Recognition (ICDAR) [2], thanks to these novel methods our system outperforms all the other competing text localization approaches on the most popular standard text localization dataset available in literature: ICDAR Challenge 2 Task 1 Robust Reading dataset. All of that while keeping an excellent computational complexity, low training times and high duplicability.

Even though the results achieved by this first text localization algorithm were encouraging, the test computational complexity of our hybrid region-based/connected component (CC)-based solution does not allow its use in real-time text spotting applications, mostly due to the time required to perform moving window analysis and visual features computation at multiple scale levels. For this very reason, we decided to focus our research activity towards the development of a more innovative deep-based system which exploits the same key ideas of our hybrid algorithm (Multi-resolution Maximally Stable Extremal Regions, Text Confidence Maps, Non-maximum Suppression, *etc.*) but performs end-to-end text spotting in natural scene using Convolutional Neural Networks and GPU processing to achieve considerably better results both in terms of detection/recognition accuracy and computational complexity.

The use of Convolutional Neural Networks for audiovisual data processing is not novel; in fact, the most recent and sucessful way of approaching difficult Computer Vision tasks (like text localization/recognition) involves deep architectures. In complex tasks like object classification on the *Large Scale Visual Recognition Challenge 2015* dataset [3] (ILSVRC2015), where each object detection algorithm has to distinguish between 1000 different object categories, it is unfeasible to compute hand-crafted visual features for each of the 1.2 million training images, and to train traditional classifiers (like Support Vector Machines, Multi-layer Perceptrons, *etc.*) to recognize between that large amount of object classes. Deep models like Convolutional Neural Networks solve this problem because they are able to automatically extract highly discriminative visual features from training data with limited tuning effort. Since Convolutional Neural Networks have been extensively used in our deep-based end-to-end text spotting work described of Chap. 4, a brief introduction of their fundamentals is provided in Chap. 2.

Similarly to our previous work on text localization, in our deep-based end-to-end text spotting algorithm work we tried to achieve good results while keeping an acceptable computational complexity. This is not a trivial task since all of the other deep-based works for end-to-end text spotting proposed in literature take weeks or even months to be trained using powerful CUDA-based graphic cards. On the other hand, our simple deep-based method, based on slight evolutions of the original LeNet Convolutional Neural Network architecture, reaches nearly human performances for some practical applications (gas flow meter reading and license plate recognition) and its results are comparable with those achieved by other more complex state-of-the-art deep-based methods

[2] http://rrc.cvc.uab.es/

on the challenging ICDAR dataset. All of this while maintaining a low computational complexity (the algorithm is roughly 10 times faster than our previous text localization method) and almost zero tuning effort.

2

Background and Related Work

In this chapter some fundamental concepts and related works that are useful to understand the methods on text localization and recognition described Chap. 3 and 4 are introduced and briefly described.

Sec. 2.1 provides an overview on how text localization and recognition tasks have been tackled by other researchers throughout the past ten years; beginning from traditional region-based approaches to newer connected component (CC)-based methods, and concluding with the most recent deep-based algorithms. For this latest class of algorithms an introduction on Convolutional Neural Networks is also provided; giving a brief overview on the fundamental elements that are typically used in deep models for Computer Vision tasks.

Sec. 2.4 provides a description of the datasets typically used in literature to evaluate and compare text localization/recognition approaches, with some visual examples of the difficult situations that need to be faced and addressed when doing text localization/recognition from natural scene images. Even though most of the datasets used in our works of Chap. 3 and 4 are standard, we have also built and manually tagged one specific dataset that has been used to evaluate the performance of text spotting algorithms in unconstrained real-world application: FlowMeter Database [4].

Unlike most of the other datasets proposed in literature (KAIST Scene Text [5], ICDAR Robust Reading [6, 7], MSRA Text Detection 500 [8], Street View Text [9], *etc.*), FlowMeter DB contains images of arbitrary rotated text with motion blur, lack of focus, gravel on the text to be recognized, and extremely wide light variations. All of these

conditions are commonly found in images aquired using mobile devices and, as shown in Chap. 4, they are poorly addressed by competing state-of-the-art approaches, which fail at correctly recognizing text from those images. On the other hand, our method successfully reach nearly human detection accuracies (2% less than human performances) for the same text spotting task.

2.1 Text Localization and Recognition

2.1.1 Region-based

Region-based approaches typically employ sliding-window classifiers to process the given image, densely analyzing all the local regions of the image looking for potential text characters of interest.

In order to reach satysfying results and detect text elements at different scales, similarly to most of the sliding-window based object detection approaches proposed in literature, the image needs to be processed in a multi-resolution manner. A pyramid of images is built by resizing the image with different scale factors; and a properly trained sliding-window classifier processes each pyramid level to detect potential text comoponents. Once the sliding-window classifier has densely analyzed one level of the pyramid, it typically generates a *text confidence map* in which the intensity level associated with each pixel denotes the probability it belongs to foreground (text) or background (noise), as in Fig. 2.1.

Text confidence maps obtained at the different pyramids levels are *stacked* (usually by summing and normalizing) together to form a final text confidence map that needs to be further processed to detect/filter the enclosing bounding boxes for the potential regions of text. This filtering procedure may be approached using a wide variety of techniques. Most of the successful region-based algorithms proposed in literature performs Non-maximum Suppression [10] (NMS) on the final overall *text confidence map* to discard overlapping or insignificant detection windows, and then stack surviving regions together to obtain the potentially relevant bounding boxes for text elements in the processed image.

NMS is a common operation in Computer Vision algorithms for object detection, the goal is to decrease the number of potential bounding boxes of interest by suppressing the ones overlapping by a certain amount. In details, the detected bounding boxes are ranked and ordered by their probability of belonging to foreground (their activation value); for each pair of bounding boxes, their overlap, defined as in [11] ($overlap(bb1, bb2) = area(intersect(bb1, bb2))/area(union(bb1, bb2))$), is computed. If the computed overlap value is greater than the set overlap threshold (in Pascal VOC Challenge and Computer Vision in general, overlap threshold is usually set at 0.5) then the bounding box with the lower score is suppressed.

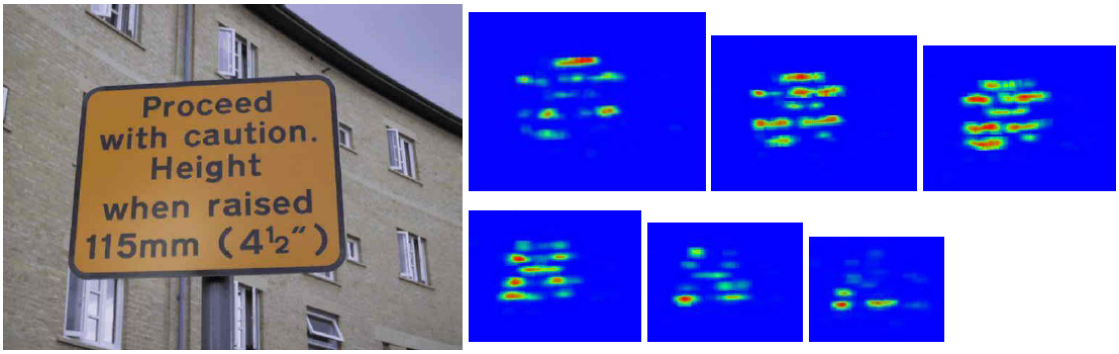


Figure 2.1: Traditional multi-resolution region-based text confidence maps (image from [12], and from the PhD thesis of L. Neumann).

Although NMS is a simple operation, it is also computationally expensive when exhaustively executed over all the potential text regions detected in the *text confidence map*: its complexity is $O(n^2)$, where n is the number of processed bounding boxes. For this very reason *greedy* algorithms are usually employed in place of classic NMS. For example, when n is very large, it is useful to split the data in half and run NMS on each part, and then combine those two parts and run once again NMS. This *greedy* algorithm may lead to a final result that differs from the one obtained using classic NMS but, considering that the analysis of an image in a dense sliding-window manner at multiple resolutions is already expensive in terms of comoutational complexity, to obtain an acceptable computanional complexity every optimization is required.

Once the potential bounding boxes for text elements of interest are found, there are different ways in which the text elements/words can be recognized. Among the most relevant region based methods proposed in literature throughout the last decade [9, 12-18] the two works that are worth mentioning due to their particular novelty over competing approaches are the ones of Pan *et al.* [12] and Wang *et al.* [9].

Pan *et al.* [12] build a text confidence map by processing images in a sliding-window manner, using Waldboost [13] and Histogram of Gradients (HOG) [14] features. The *text confidence map* is used, together with other geometric features and a properly trained Multi-layer Perceptron [15], to compute the binary and the unary weights of a component neighborhood graph built over a set of connected components extracted using Niblack's text binarization algorithm [16]. Conditional Random Fields [17] (CRF) are used to filter out non-text components from the graph, while the remaining neighboring elements are clustered together into Minimum Spanning Trees [18] (MST) to form text words.

In our approach of Chap. 3, we also exploit a similar text confidence map to identify potential regions of text, but we do not use all these complex and hard to reproduce clustering algorithms or CRF to select regions of text from the processsed image; instead we exploit the fundamental concepts of CC-based approaches (described in the Sec. 2.1.2)

to effectively and efficiently detect the bounding boxes for text components on interest. The effectiveness of our hybrid method (because it exploits both the concepts of region-based and CC-based approaches) is proved by the state-of-the-art results it achieves for ICDAR 2013 Challenge 1 Task 2, where it beats all the other published and unpublished competing approaches proposed in literature at detecting text from natural scene images.

The second algorithm that it is worth analyzing is the one of Wang *et al.* [9] which performs end-to-end text recognition using Random Ferns and Pictorial Structures. Even though the pipeline of this method does not have any particular novelty, the part of the work that is particulary interesting and novel is the choice of using synthetic positive training data: roughly 1000 images are synthesized per text character using 40 different fonts, adding Gaussian noise and applying random affine deformations (similarly to the work of [19], which will be discussed in Sec. 2.1.2). It is interesting to see how the classifier trained exclusively using synthetic positive data achieves the same F-measure of a Nearest-neighbour classifier [20] (NN) trained with Histogram of Oriented Gradients [14] (HOG) features extracted from native data.

Another interesting idea from [9] is the choice of extracting negative training samples from classes of Microsoft Research Cambridge Object Recognition Image Database [21] (MSRC): classes like *buildings* and *countryside* deeply resemble the background patterns of ICDAR images and help in reducing the number of false-positive errors produced by the sliding-window classifier. Training region-based methods using synthetic data has become quite popular during the last years, because it is very difficult to gather a satisfying amount of correctly tagged training images that lead to acceptable results in real-world applications when used to train a traditional sliding-window classifier. The use of syntetic data also skyrocketed with the introduction of deep-based text spotting algorithms that require millions of examples to reach decent detection rates. For example, the performances of PhotoOCR [22] increase by more than 10% when adding millions of syntetically generated training examples to the positive training set. Unfortunately, researchers have yet to present a method for generating an infinite amount of syntethic images which satisfactorily simulate natural scene images.

Region-based methods have become less popular with the introduction of novel CC-based approaches that overcome most of the limitations imposed by sliding-window classifiers: (i) high computational complexity, especially during feature computation at multiple scales; (ii) false-positive detection errors, as some local regions in natural images are virtually undisguisable from text components [23] (e.g. the corners of windows may be seen a T or $+$ when locally analyizing that particular portion of image); (iii) long training times; (iv) and low portability to mobile enviroments, to name a few.

While most of the aforementioned problems could be solved using novel Computer Vision techniques, such as Approximated Features [2], Aggregate Channel Features [24], *etc.*, region-based methods slowly became less popular for text localization/recognition in

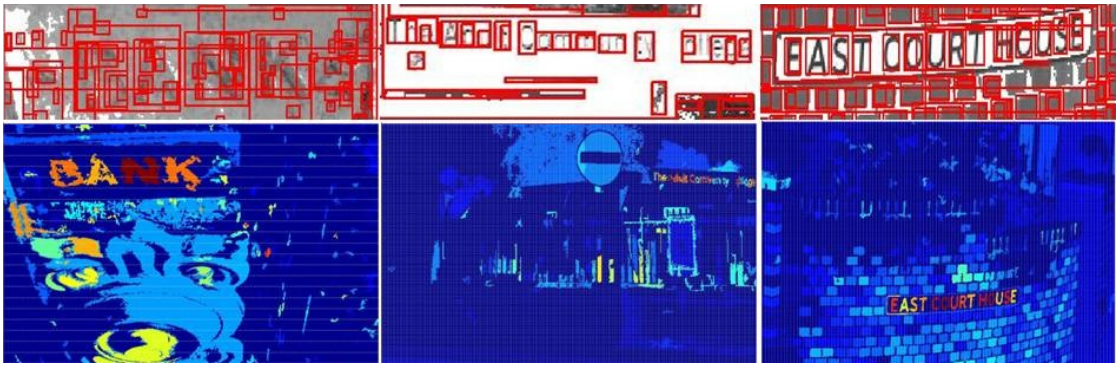


Figure 2.2: CNN filtered Maximally Stable Extremal Regions (image from [27]). 1st row: enclosing bounding boxes for the detected Maximally Stable Extremal Regions; 2nd row: intensity maps for text-likelihood values assigned to each Maximally Stable Extremal Region by the CNN.

natural scene images, first replaced by CC-based methods, and then by more innovative deep-based approaches which are much faster and require less tuning to reach optimal results. In fact, during the last years the effort of Computer Vision researchers shifted from "finding the best classifier and features for the given task" [9, 12, 24, 25], to "finding the training data that leads to best results when used to train a deep model" [3, 4, 26, 27, 28]. Thus making traditional text localization/recognition region-based approaches, whose performances are highly related to the features used for classification, obsolete.

2.1.2 Connected Component (CC)-based

The main intuition behind connected components (CC)-based approaches is that text characters usually show uniform geometric characteristics. For example, the color of a text character stays the same for the whole digit/letter, the stroke width of the boundaries for the character has a slow intra-variation within the character itself, text characters are usually placed on a high contrast background to increase readability, and can be clustered on a same line to form words of text. All of these geometric properties have been exploited by researches over the years to propose text localization/recognition methods that identify and extract text character as stable connected components from the processed images.

More in details, most CC-based text localization and recognition methods [25, 29, 30, 31, 32] either exploit Maximally Stable Extremal Region [33] (MSER) to identify potential text components that are filtered and clustered together to form words, or enhance/exploit the Stroke Width Transform [23] (SWT) algorithm to identify connected components having low intra-stroke variance [8, 34, 35].

MSER is a method for blob detection in images to compute for a given gray image

a number of co-variant regions called MSER. A MSER is a stable connected component of some gray-level sets of the image, where the concept of *stability* is defined as a region that stays nearly the same through a wide range of thresholds.

More specifically, when thresholding an image at different intensity levels it is possible to compute, for each threshold level, the geometric properties of the connected components in that level. Once the threshold level is changed it is also possible to compute the variation between two related connected components at the different thresholds. If the computed variation (in terms of area, perimeter, *etc.*) is small enough for a sufficiently large amount of thresholds, the connected component is considered *stable* and it is identified as MSER. The word extremal refers to the property that all pixels inside the MSER have either higher (bright extremal regions) or lower (dark extremal regions) intensity than all the pixels on its outer boundary.

Similarly to SIFT [36], MSER features can be used as keypoint descriptor for images, showing great results when paired with other keypoint descriptors [37]. In text detection, due to the geometric properties of text characters, MSER have become quite popular during the last years. Unfortunately, classic MSER cannot capture text characters that are blurred, unfocused, *etc.* For this very reason, many researchers tried to increase the coverage rates (the percentage of ground-truth text elements identified as stable components) of MSER by proposing alternative algorithms for extracting connected components which are specifically designed for text elements. For example, Neumann *et al.* [38] proved that extracting Extremal Regions (ER) (CC that are not maximally stable) from multiple image channels (RGB $\cup$ HSI $\cup$ ∇) leads to a coverage rate of 95% over ground-truth character annotations for ICDAR 2011 dataset. This is an outstanding coverage result for ICDAR 2011, but the amount of extracted ER is very high when compared to the amount MSER (roughly 100 times higher), thus requiring the algorithm of [38] to use computationally incremental features (computed in constant time $O(1)$ for every ER) to maintain an acceptable computational complexity. Those so-called computationally incremental features are not discriminative enough to allow the algorithm to reach state-of-the-art text localization results.

Other more efficient variants of the original MSER algorithm have also been proposed: (i) Multi-channel MSER [32] (M-CHN); (ii) Multi-resolution MSER [37] (MR-MSER); (iii) Edge-preserving MSER [25] (eMSER); (iv) and Edge-enhanced MSER [39] (EE-MSER), to name a few. More in details, in the work of Li *et al.* [25] Edge-preserving MSER are obtained by extracting MSER from images incorporating gradient magnitude and intensity channels information. eMSER are robust to blur and therefore overcome some of the limitations of traditional MSER (similarly to Edge-enhanced MSER [39]).

Another technique for improving the coverage rate of MSER over text elements is the one proposed by Forssén and Lowe [37]: a pyramid of images is built and Multi-resolution MSER (MR-MSER) are extracted at multiple scales (1 scale per octave). This

multi-resolution approach 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. Even though MR-MSER have never been used before for text localization from scene images, in our works of Chap. 3 and 4, we prove that they can be combined with the multiple channel technique of [25, 29, 38] to extract entire words of text from natural images.

SWT [23] is another algorithm that is quite popular among text localization works. Unlike MSER-based algorithms, SWT does not threshold the image at multiple levels to look for stable connected components. Instead it looks for edges in the image (using Canny Edge Detector algorithm [40]) and builds a stroke width map in which the value assigned to each pixel denotes the width of the edge it belongs to. The stroke width map is built by following the direction of the gradient on every edge until another edge is found. The values of the pixels lying on the direction of the gradients is incremented by the length of the segment between the two edges. Given the stroke width map, connected components sharing similar activation values (similar stroke width) are identified as potential text characters.

Unfortunately, the implemetation of the original SWT algorithm has not been released to public yet, and the published results of the algorithms are difficult to reproduce due to the lack of implementation details in the reference paper. In some works, generic object window proposal/detection methods (Edge Box [41], Selective Search [42], ACF [24], *etc.*) have also been used in place of MSER variants for bounding box generation for text elements in natural scene images [43].

Lately, object window proposal algorithms have become quite popular in Computer Vision. The first work that introduced the concept of object as a generic entity having homogeneous visual characteristics (high contrast, boundary continuity, visual saliency, etc.) is the Objectness algorithm (originally proposed by Alexe *et al.* [44] and then extended in [42, 45, 46, 47]) which generates a set of ranked object window proposals for every given image, where the rank level denotes the probability that the window proposal encloses an object.

In text localization algorithms, those window proposals are processed to analyze whether they correspond to the bounding boxes of text elements or not. Introducing object window proposals algorithm to the task text localization may be a good idea as objectness-like methods have been sucessfully used as base detectors by many recent state-of-the-art object segmentation/detection/localization methods [48, 49, 50] and can significantly lower the computational complexity of the detection method. In our experiments of Chap 3 and 4 we provide a comparison between most of the variants of MSER proposed in literature throughout the years and the most recent object window proposal algorithms. Our results show that MSER variants are always better at cap-

turing text characters than generic object window proposal algorithms, both in term of provided coverage rate and computational complexity. We also show how our "proposal augmentation technique" (see Chap. 4, Fig. 4.2) can significantly boost detection recall (number of ground-truth text character annotations covered by at least one generated proposal) for all the evaluated datasets.

Once connected components have been extracted from the given image, the pipeline of CC-based methods is usually straightforward: (i) visual features are computed for each connected component; (ii) they are processed by a properly trained classifier that determines whether they contain text elements of interest or not; (iii) finally, connected components sharing similar visual characteristic are grouped together to form text lines and words.

Unlike region-based approaches, CC-based algorithms are efficient to train and test. Some CC-based methods can even process a stream of images from webcam in real-time on a desktop machine (see the implementation of [38] provided in OpenCV 3.0).

However, their recall values on ICDAR datasets are typically low, because uncommon text characters and fonts (company logos, graffiti, unfocused characters, *etc.*) are usually discarded due to their irregular geometric features.

In our work for text localization of Chap 3 we overcome this problem by proposing a hybrid algorithm that does not discard connected components based on some visual geometric properties; instead it exploits the *text confidence maps* generated by an Aggregate Channel Feature (ACF)-based sliding-window classifier. Thanks to this choice, our algorithm can overcome both the limitations of region-based and CC-based approaches, reaching state-of-the art results for ICDAR 2013 (especially in term of recall, which is 20% higher than the second ranked algorithm).

2.1.3 Deep-based

Convolutional Neural Networks [51] (CNN) have recently been sucessfully used for text localization and recognition in natural scene images. Since deep-based methods are very recent, there are not many deep-based text spotting works in literature. In this section, I discuss the three most relevant works that have been proposed during the last years.

In [19], Coates *et al.* realized the discriminative power of features unsupervisingly learnt by a CNN at identifying text characters from natural images. However, they could not find a proper way to identify bounding boxes for text elements. Therefore they did not participate in the ICDAR challenge to evaluate how their features performed when evaluated using ICDAR's evaluation protocol. Instead, they evaluated the Precision/Recall of their deep-based model using per-pixel based accuracies computed from the ground-truth annotations for ICDAR 2003 images, thus making it almost impossible for other researchers to compare existing results in literature with theirs.

In [28], a single very large CNN has been used for integrated text localization and

Figure 2.3: Edges are detected by convolving the image with the kernel of Sec. 2.2.

recognition of Street View House Numbers [19] (SVHN) and CAPTCHA, thus removing the need for using local windows or proposals as in region-based and CC-based methods.

While this integrated end-to-end text localization and recognition approach seems promising, since it reaches nearly human detection rates on Google SVHN and CAPTCHA datasets; it can only be applied for the localization/recognition of text sequences whose length is known a priori, and a large amount (10 million or more manually tagged samples) of training data is required to obtain acceptable results.

Moreover, the proposed CNN model requires weeks to be trained using DistBelief [52] (a powerful large scale distributed deep network architecture of Google) and it is therefore difficult/impossible to reproduce and re-train for other text related tasks.

In [43], multiple very large CNN trained solely on synthetic data are used to localize and read text word proposals from Edge Box and ACF detector.

This latest approach is similar to the one we propose in Chap. 4 however, we work at text character level using augmented MR-MSER proposals in place of synthetically generated training data.

Moreover, unlike most of the other deep-based methods proposed in literature, in our deep-based algorithm described of Chap. 4 we pay particular attention to the reproducibility of the method on common desktop machines, employing only small variants of LeNet CNN [53]. Despite the simplicity of the CNN used in our work, the text spotting results we obtain are comparable with human performances on a real-world application (FlowMeter DB dataset) and also with competing traditional state-of-the-art methods on ICDAR 2015 dataset.

2.2 Convolutional Neural Networks

Convolutional Neural Networks [51] (CNN) broke state-of-the-art results for several Computer Vision tasks, and are now the dominant networks used for feature extrac-

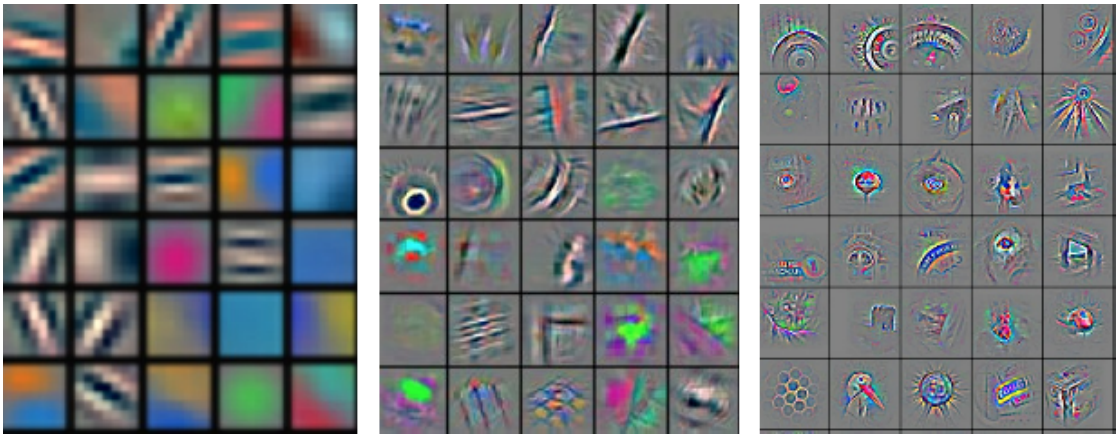


Figure 2.4: Feature visualization of the kernel weights learnt by AlexNet Convolutional Neural Network [54] on ImageNet dataset [3, 51]. In the first convolutional levels (left and middle) simple features like edges and gradients are learnt; at deeeper levels (right) the network starts to learn more complex features like eyes, noses, flowers, *etc.*

tion from audiovisual and textual data. [1]

CNN are slight variations of traditional Feed-forward Neural Networks. They take biological inspiration from small regions of cells in the visual cortex that are sensitive to subregions of the visual field. In Machine Learning, these regions of cells are referred to as a receptive fields.

Receptive Fields are implemented in the form of weighted matrices, referred to as *kernels*; which, similarly to their biologically inspired counterparts, are sensitive to similar local subregions of an image.

The degree of similarity between a subregion of an image and a kernel may be computed simply convolving the subregion using the given kernel. For example, a simple 3×3 kernel like:

$$\begin{pmatrix} -5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

may be convolved over one given image to detect and inform about the presence of edges as in Fig. 2.3.

When used in CNN, kernel weights may be learnt from the training data to extract basic features like edges, gradients and blurs; and when the network has enough convolutional layers, the kernels will start learning feature combinations off of the previous layers. From simple features like edges, gradients and blurs, kernels start to learn more complex features that are highly discriminative for the processed images.

[1] Are we there yet? - http://rodrigob.github.io/are_we_there_yet/build/

For the ImageNet Challenge [3] those features may detect the presence of eyes, noses, hair, *etc.* Similarly to the ones learnt by one of the top performing CNN for image classification on ImageNet Challenge: AlexNet [54] (Fig. 2.4).

Traditional works on audiovisual and textual processing required researchers to train supervised models using some kind of features extracted from the given training data. This is an expensive and fragile task, and it is difficult sometimes to understand if the feature being extracted is discriminative enough for the task to be solved.

For example, the manuscript of Vondrick *et al.* [55] contains an analysis of the discriminative power of HOG features; the paper provides an approximated inversion of the HOG function and tries to understand why even the best classifiers are making some strange mistakes for apparently simple images from Pascal VOC Challenge dataset [11]. For example, in the image of Fig. 2.5a there is a strange and apparently unexplainable car detection window in the middle of the water even though there is clearly no car in the water in the original image. The mistery is solved when looking at the inverse visualization of the HOG feature for that local portion of the image: the car is hidden in the HOG descriptor (Fig. 2.5b).[2]

This shows that using just HOG features no classifier would be able to avoid that car false-detection, no matter how much time is spent for the training phase and no matter which classifier is being trained.

On the other hand, a CNN does feature extraction on its own. By convolving the given image a sufficient number of times it is possible to let the network, with minimal preprocessing of the data, create/detect the set of visual features that work best for the specific domain. In most cases, self-learned CNN features work better than most of the algorithmically and hand-crafted features.

CNN are composed of building blocks corresponding to linear and non-linear operators. As previously described, the most common operator in a CNN is the regular linear convolution by a filter bank. Given an image x with k channels, and a kernel w, the convolution operator generates another image y in the following way:

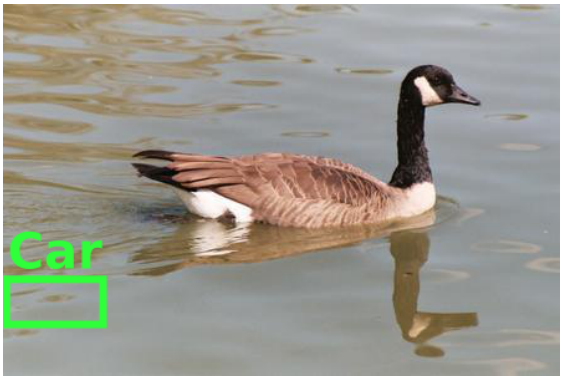
$$y_{i'j'k'} = \sum_{ijk} w_{ijkk'} x_{i+i',j+j',k} \tag{2.1}$$

where k' corresponds to the number of filters/kernels in the convolution.

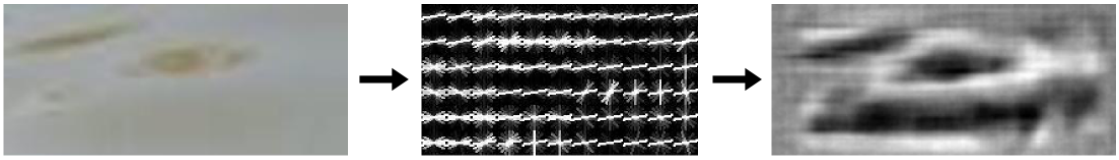
In addition to the previously described linear convolution, there are several non-linear operators involved in the classification process. The most common non-linear operators that can be found in CNN are: Non-linear Gating, Pooling and Normalization.

The simplest non-linearity can be obtained by following a linear filter by a Non-linear Gating Function applied identically to each component of a feature map. The simplest

[2]HOGgles: Visualizing Object Detection Features - http://web.mit.edu/vondrick/ihog/



(a) False-positive car detection.



(b) Visualization of the HOG descriptor for the false-positive detection region.

Figure 2.5: The mistery behind the false-positive car detection in the image (a) is solved when looking at the visualization of the HOG descriptor for the false-positive patch (b). The trained model misclassifies that local part of the image because the orientation of the gradients in the computed HOG descriptor is very similar to the ones of a car [55].

Non-linear Gating Function is the Rectified Linear Unit (ReLU), which is defined as:

$$y_{ijk} = \max\{0, x_{ijk}\} \tag{2.2}$$

For every element position x_{ijk} the function takes the maximum value between 0 and x_{ijk}. While this may seem like a useless operation, setting many of the values in a feature map to 0 substantially improves CNN training times, since the derivative of 0 is constant and can be computed in constant time.

One of the other commonly used non-linear operators in CNN is Pooling. Pooling operates on individual feature channels, fusing nearby feature values into one by the application of a suitable operator. The most common choice is Max-pooling, which is defined as:

$$y_{ijk} = \max\{y_{i'j'k} : i \leq i' < i + p, j \leq j' < j + p\} \tag{2.3}$$

where p denotes the size of the $p \times p$ Max-pooling window. Max-pooling creates position invariance over larger local regions and down-samples the input image by a factor of $p \times p$

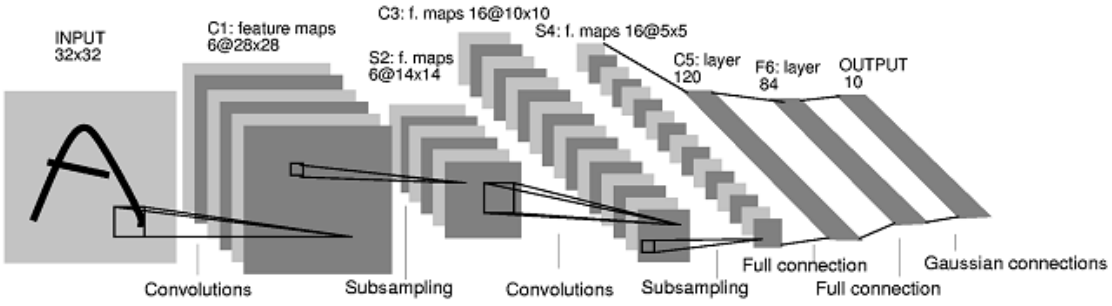


Figure 2.6: Visual representation of the architecture of one of the first Convolutional Neural Networks sucessfully used for feature extraction from visual data: LeNet [53]:

along each direction. It leads to faster convergence rates by selecting superior invariant features which improve generalization performances.

Another non-linear operator that may be used in CNN is channel-wise Normalization which is used to normalize the feature channels at each location in the feature map x. The operator is defined as:

$$y_{ijk'} = \frac{x_{ijk}}{\left(\kappa + \alpha \sum_{k \in G(k')} x_{ijk}^2\right)^{\beta}} \tag{2.4}$$

where κ, α and β are the parameters for the normalization. Channel-wise Normalization is also crucial to make the network invariant to small visual changes in the training samples.

CNN are usually trained using Stochastic Gradient Descent, which is a gradient descent optimization algorithm for minimizing the error for a given objective function.

A great speed-up in training times may be obtained when training CNN by exploiting parallel execution provided by modern graphic cards. However, due to the limited amount of memory resources available on GPU, the whole set of training samples cannot be usually stored in graphic memory at training time: the samples need to be split into multiple randomly created mini-batches. The error between the predictions of the CNN and the objective function is computed for each mini-batch, and the weights of the kernels in the network are updated according to that error.

The training process is therefore composed by many iterations (processing of single mini-batches) and many epochs (processing of the whole training samples). The error on validation set is usually computed after each training epoch.

There are many open source libraries available online for the creation and training of CNN. In our work of Chap. 4 we have used Vedaldi's VLFeat MatConvNet [56] to design and test our CNN in Matlab 2014a, and Berkeley Caffe [57] to deploy the trained network models in CUDA/C++.

To be able to train our CNN in acceptable times we have used two NVIDIA GeForce GTX 980 and NVIDIA Cuda Deep Neural Network (CuDNN), which is a library specifically developed by NVIDIA to speed up the convolution of images on CUDA-based GPU.

2.3 Fast Feature Pyramids and Aggregated Channel Features

The concept of Fast Feature Pyramids, introduced by Dollár *et al.* [2] and further inspected in [58], revolutionized multi-scale sliding window approaches by showing that image features 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 [59, 60, 61].

The work of Mathias *et al.* [59] shows that, without specific modification, the Integral Channel Feature Classifier (ChnFtrs) [24], originally introduced for pedestrian detection, can be applied to the task of traffic sign recognition to reach state-of-the-art performance; this is particularly interesting as ChnFtrs has been extended in [2] by replacing the explicit computation of multi-scale features with Fast Feature Pyramids.

The resulting algorithm, called FPDW [2] (aka *"The Fastest Pedestrian Detector in the West"*) reaches the detection rate of ChnFtrs while being 2 order of magnitude faster than competing methods. In [58], FPDW is further enhanced by introducing the concept of Aggregated Channel Features (ACF).

Following the insight of [59] and the analysis of [62] on how to build the best classifier for rigid object recognition, in the work on text localization of Chap. 3 we use an ACF based classifier to perform text localization from natural images, obtaining excellent detection rates and fast recognition times.

2.4 Datasets

2.4.1 ICDAR Robust Reading

The International Conference of Document Analysis and Recognition (ICDAR) is the premier international conference for researchers in the document analysis community for identifying, encouraging and exchanging ideas on the state-of-the-art technology in document analysis, understanding, retrieval, and performance evaluation. The term document in the context of ICDAR encompasses a broad range of documents from historical forms such as palm leaves and papyrus to traditional documents and modern multimedia documents.

The so-called "robust reading" task refers to the research area that deals with the interpretation of written communication in unconstrained settings. Robust reading is

Figure 2.7: Samples from ICDAR 2015 Challenge 2 Task 1 dataset [63].

linked to the detection and recognition of textual information in scene images, but in the wider sense it refers to techniques and methodologies that have been developed specifically for text containers other than scanned paper documents, and include born-digital images and videos to mention a few.

Over the years, ICDAR have proposed multiple manually tagged datasets and also strict online evaluation protocols. This has led to the acceptance of ICDAR's robust reading competition framework by researchers worldwide as the defacto standard for evaluation, and has promoted good practice in the field [6, 7].

Examples of images from ICDAR datasets are provided in Fig. 2.7; more examples are also provided in our works of Chap. 3 and Chap. 4. An online comparison between all the competing text localization and reading algorithms proposed in literature over the years (including our work: IWRR2014 [64]) is publicly available on the conference website. [3]

2.4.2 License Plate Recognition

License plate localization and reading from natural images is an interesting task that finds application in many real-world problems.

[3] http://rrc.cvc.uab.es/?ch=2&com=evaluation

Figure 2.8: Some examples of images from "UCSD/Calit2 Car License Plate, Make and Model Database" (1st row), "Zemris Car Database" (2nd row), and "Medialab License Plate Recognition" (3rd row). Our algorithm of Chap. 4 was trained on the first two datasets and tested on "Medialab License Plate Recognition" dataset.

Due to the privacy issues associated with car license plate numbers it is hard to find tagged publicly available datasets. The most commonly used datasets to evaluate the performances of license plate reading methods are the "UCSD/Calit2 Car License Plate, Make and Model Database", "Zemris Car Database", and "Medialab License Plate Recognition" (Medialab LPR).

Unfortunately none of these datasets are tagged for localization; therefore, to evaluate the end-to-end text localization and reading performances of our work of Chap. 4, we had to manually tag 290 images from UCSD dataset, 503 images from Zemris, and 680 images from MediaLab LPR. We then compared the performances of our deep-based method with other similar algorithms proposed in literature [65, 66]; results are presented in Tab. 4.3: the proposed method outperforms all the other approaches while also being faster than all of them.

Some examples of images from the 3 license plate recognition datasets are provided in Fig. 2.8; our method was trained using just the samples extracted from UCSD and Zemris datasets and tested using all the 680 images from Medialab LPR.

Figure 2.9: Samples from FlowMeter Database including partially occluded meters, gravel on the digits, reflections, and excessive distance from the camera.

2.4.3 Gas Flow Meter Reading

FlowMeter Database [4] (FlowMeter DB) is a text localization and recognition dataset that we manually created to evaluate the performances of our algorithm of Chap. 4 on a real-world problem: the automatic reading of gas flow meters.

The problem of reading gas flow meters is interesting because companies invest millions on human employees that roam from house to house to read the values associated with every gas meter, and working as a gas meter reader is one of the top-10 worst job of 2012 and 2013. [4] [5]

FlowMeter DB contains 6050 train and 168910 test scene images of gas flow meters. All the images were acquired using smart phones, and typically contain non-horizontal flow meters as well as difficult light conditions, lack of focus, motion blur, reflections, gravel on the digits, *etc.*; they all have been manually tagged using MIT LabelMe dataset creation interface [67] paired with Amazon Mechanical Turk. [6]

Some examples are provided in Fig. 2.9. Unlike common text spotting datasets like ICDAR, the goal on FlowMeter DB is to localize and read only the ciphers of the

[4] http://www.careercast.com/content/10-worst-jobs-2012-7-meter-reader
[5] http://www.careercast.com/slide/worst-jobs-2013-7-meter-reader
[6] http://labelme2.csail.mit.edu/

meter in the image and not the whole text. As such, the evaluation metric we used to measure the performance of our algorithm is the Sequence Transcription Accuracy metric of [28], which is defined as the rate of test images for which the predicted sequence of numbers/letters matches the respective ground-truth data.

An online demo of the approach presented in Chap.4 for some images from FlowMeter DB is publicly available online. [7]

[7] http://gasmeterreader.dista.uninsubria.it/

3

Text Localization with Fast Feature Pyramids and MR-MSER

This chapter contains an overview of our article *"Text Localization based on Fast Feature Pyramids and Multi-resolution Maximally Stable Extremal Regions"* [64] presented at the 1st International Workshop on Robust Reading (IWRR2014), in conjunction with the 12th Asian Conference on Computer Vision (ACCV2014).

3.1 Summary

In [64], we focus our attention toward the task of text localization, proposing 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 text localization method to identify uncommon text elements (*e.g.* company logos, graffiti, *etc.*) 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.

In Sec. 3.2 we introduce the motivations that lead to the development of the proposed method, which is deeply described in Sec. 3.3; in Sec. 3.4 we present the text localization results it achieves for ICDAR2003 and ICDAR2013 Challenge 2 Task 1; an experimetal

evaluation of the method is provided in Sec. 3.4.

3.2 Introduction and Motivations

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.* [12], algorithms for text localization can be classified as either region-based [9, 12, 19, 68] or connected component CC-based [25, 29, 30, 31, 32]. 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.

As previously stated in chapter 2, 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 [23].

Most CC-based text localization methods [25, 29, 30, 31, 32] identify stable connected components using Maximally Stable Extremal Regions (MSER) [33]. 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 2013 [7] competition successfully detect blurred or uncommon (graffiti, company logos, *etc.*) text characters, as those text 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 [58] with Multi-resolution Maximally Stable Extremal Regions (MR-MSER) [37] to propose an 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 [58].

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.

Figure 3.1: Examples of uncommon and difficult text components successfully detected by our method (images from ICDAR 2003 and ICDAR 2013 datasets). These uncommon text fonts are usually not recognized by competing text localization approaches, as they do not satisfy the strict geometric requirements needed to be classified as text elements by competing CC-based methods.

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 2013 [1] (some examples are provided in Fig. 3.1), despite its simplicity, the proposed approach succeeds where competing CC-based text localization methods usually fail, and achieves good results for ICDAR Challenge 2 Task 1 Robust Reading dataset.

3.3 Algorithm

The proposed approach is presented in this section: a binary classifier based on Fast Feature Pyramids (Sec. 2.3) and Aggregated Channel Features (Sec. 3.3.1) is trained using natural, synthetic and semi-synthetic data collected from multiple datasets and/or artificially generated (Sec. 3.3.2); predictions from the classifier are used to build a text

[1] http://dag.cvc.uab.es/icdar2013competition/?ch=2&com=results, **Method:** IWRR2014.

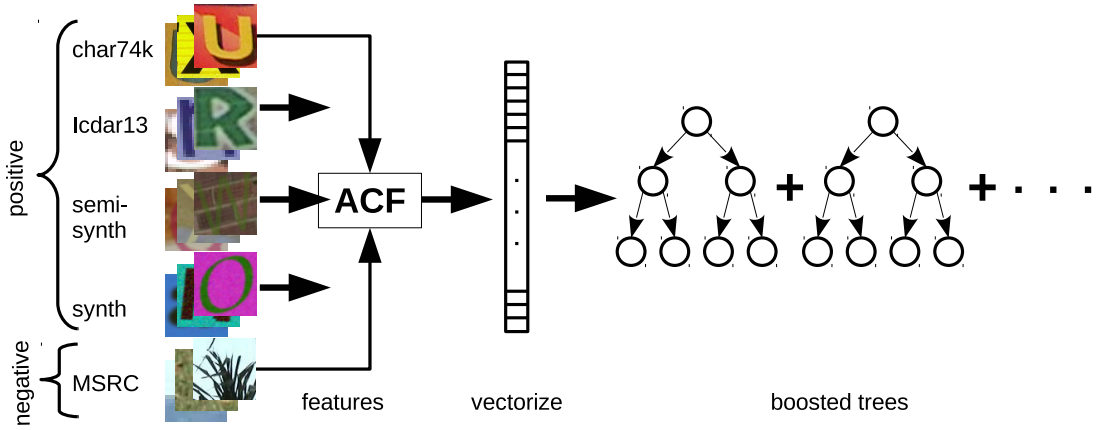


Figure 3.2: Aggregated Channel Features (ACF) extracted from negative samples from MSRC and positive natural, synthetic and semi-synthetic samples from different datasets (ICDAR 2003/2013 [6, 7], Char74k [69], and Synth [9]) are used to train boosted depth-two decision trees (also known as *stump classifiers*).

confidence map in which potential regions of text are highlighted (Sec. 3.3.3); the text confidence map is used, together with MR-MSER (Sec. 3.3.4), to identify potential bounding boxes for lines of text in the processed image (Sec. 3.3.5).

An analysis of the computational complexity of the proposed approach and implementation details are provided in Sec. 3.3.6 and Table 3.1.

3.3.1 Text Region Detector

The first step in our text localization pipeline is to build a text confidence map by detecting potential regions of text using a multi-scale sliding window ACF detector [58].

ACF uses Aggregated Channel Features, which are extracted by smoothing the processed image I with a [1 2 1]/4 filter and then computing 10 different channels: normalized gradient magnitude, histogram of oriented gradients (6 orientations) and LUV.

In our implementation, the channels are condensed into 4×4 blocks and once again smoothed using the same approximated Gaussian kernel before being concatenated together to form single descriptors.

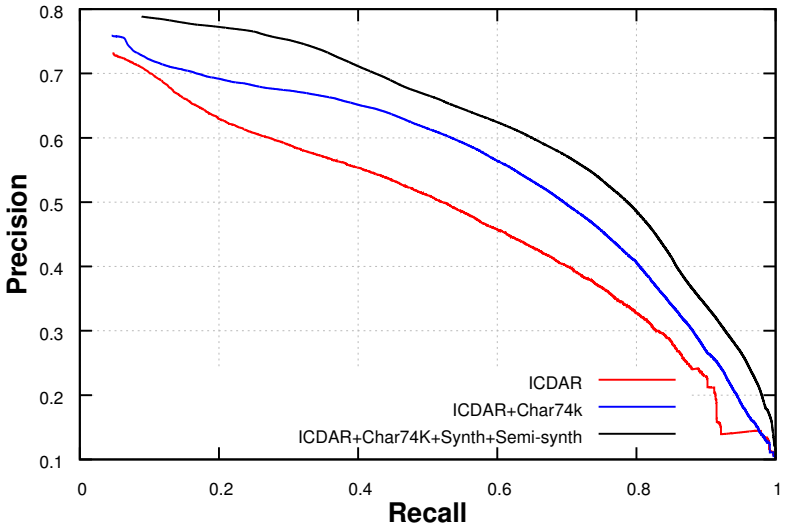
We tune the ACF classifier to reach optimal detection rates for text detection from scene images by setting the sliding window size to 32×32 pixels and the window stride to 16 pixels both in the horizontal and vertical directions.

To deal with the large variation in size of text components in ICDAR datasets, 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.

(a) ICDAR'13 [7] examples.

(b) Synth [9] examples.

(c) Semi-synth examples.



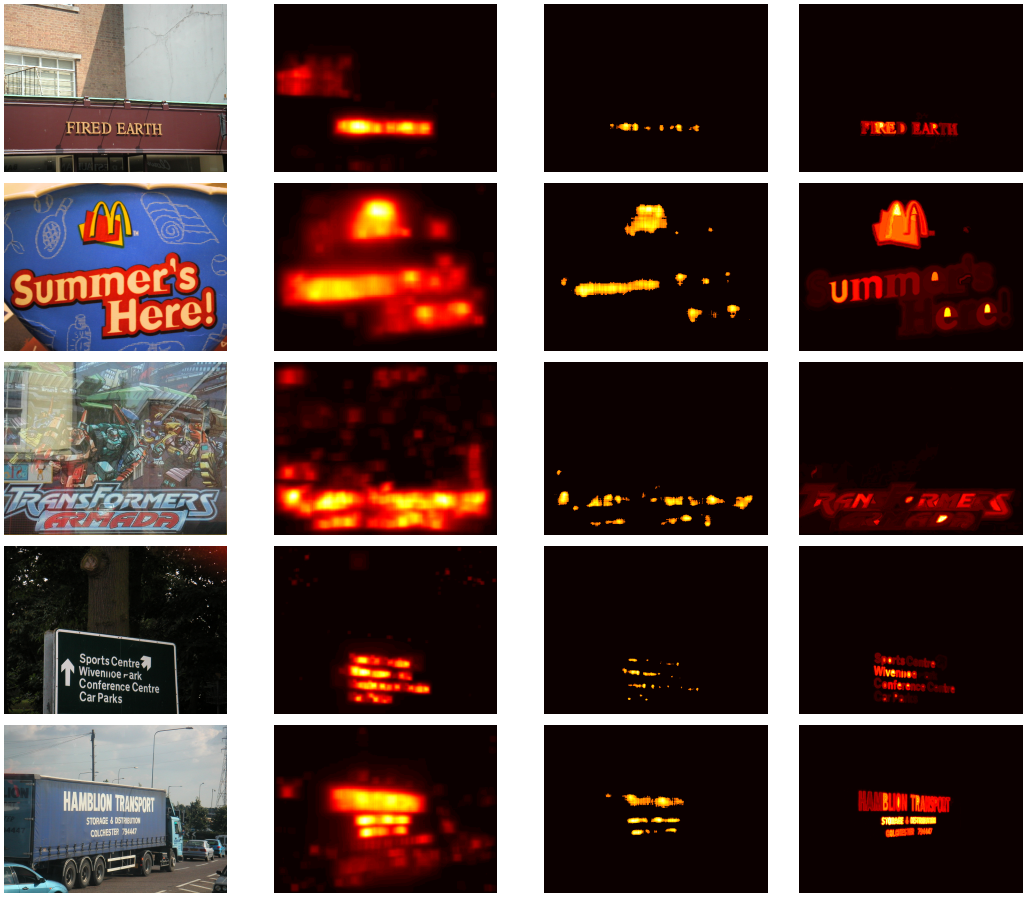
(d) Precision/Recall (PR) curves for the trained ACF based classifier.

Figure 3.3: Augmenting the positive training set with synthetic and semi-synthetic data increases the detection rate of the approximated feature based classifier. Semi-synthetic samples are generated by placing random sized artificial characters in random positions over images from MSRC dataset [21]; random jitter (translation and rotation) is applied to increase the robustness of the classifier.

For each octave, 7 scales out of 8 are approximated using the λ coefficients [58] inferred from 1000 samples randomly extracted 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 because tiny text components are not correctly detected The same behaviour when removing low pyramid levels or when improperly altering the size of the sliding window.

Our ACF classifier is composed by a discrete AdaBoost of 2048 depth-two decision trees (Fig. 3.2) that is trained using the speed-up technique of Appeal *et al.* [70], which



(a) source (b) text conf. (*tcm*) (c) thr. & NMS (d) textness (*tex*)

Figure 3.4: True-positive regions discarded when thresholding the text confidence map *tcm* (b-c) are recovered in the textness map *tex* using MR-MSER (d).

evaluates the discriminative power of features on a subset of the training data and uses that information to prune the underachieving ones throughout the training process.

As shown in [71], the performance of binary boosted classifiers can be improved by bootstrapping the training data: 2 or 3 bootstrapping iterations can increase detection rates by almost 10%. We perform 3 rounds of training, each time increasing the number of boosted weak learners (512/1024/2048); at each training 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 the amount of false-positive errors generated in the subsequent round.

Similarly to [58, 61] and unlike [71], false-negative samples are not bootstrapped because text components wrongly classified as background are recovered using MR-MSER, as described in Sec. 3.3.5.

Thanks to the training routine of [70], even using 50000 positive/negative samples and 3 rounds of bootstrapping, the ACF classifier can be fully trained in less than 3 minutes on a Intel Core i5 (see Table 3.1).

3.3.2 Training Data

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

Considering that state-of-the-art results have been obtained in rigid object recognition by methods based on ICF and ACF [58, 59], we assume that good results may also be obtained in text detection using the same set of features if a decent amount of 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 [21] (*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.

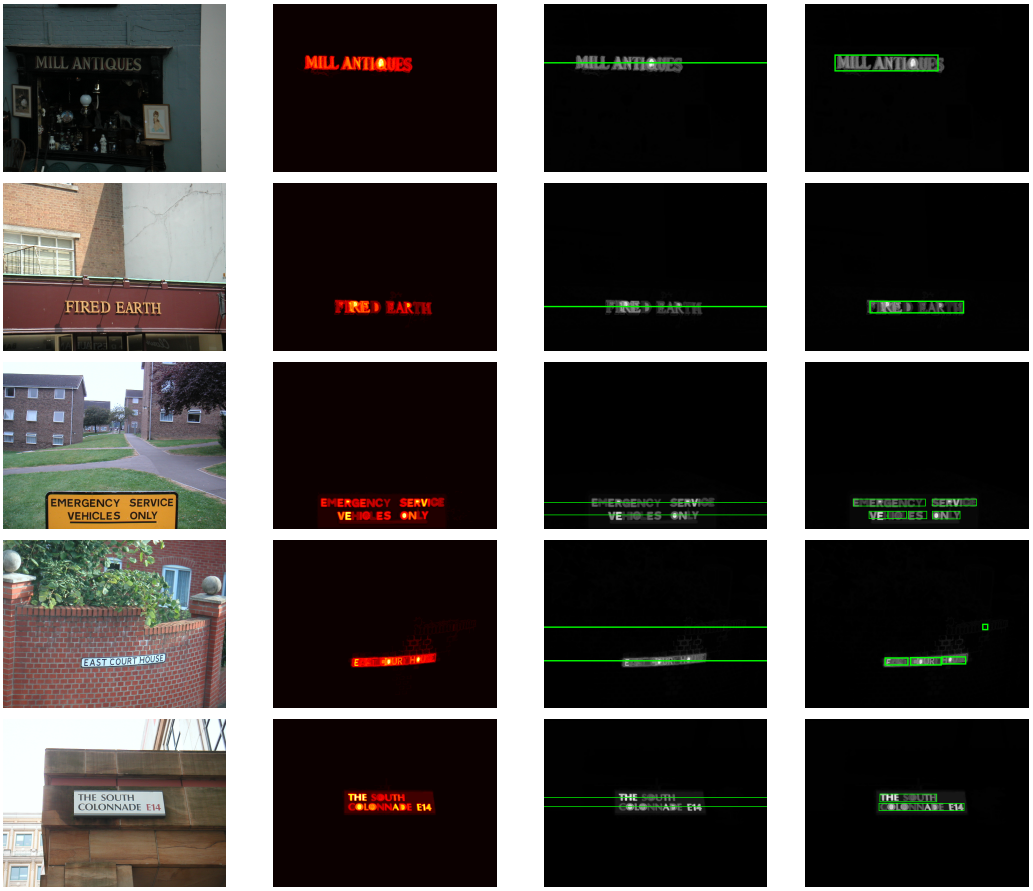
Extracting negative examples at multiple scales reduces the number of false-positive errors generated at low octaves in the feature pyramid.

Gathering positive training samples is a challenging task, poor results were obtained when training our classifier using just the ≈5400 samples from ICDAR'13 [7] (see Fig. 3.3).

For this very reason, we augmented the set of positive training data with: ≈8000 images from the *GoodImg* class of Char74k English dataset [69], ≈6200 artificial images from the publicly available Synth dataset [9] (vertically cropped to remove neighboring characters) and ≈30000 semi-synthetic samples obtained by combining natural data from MSRC with synthetic fonts.

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 character and its surroundings, we compute the histogram of the patch on which each character is pasted and discard samples that are human unreadable (zero contrast between character and background).



(a) source (b) text conf. (*tcm*) (c) thr. & NMS (d) detection (*tex*)

Figure 3.5: Text line formulation algorithm pipeline: the text confidence map (b) is thresholded (c); words are identified using both textness map *tex* and MR-MSER (d); final components are grouped together via Mean-shift [72].

Fig. 3.3 shows how the positive sample sets we aggregate complement each others: samples from ICDAR and Char74k (Fig. 3.3a) contain uncommon and handmade characters that cannot be artificially generated; the synthetic data from Synth (Fig. 3.3b) is useful to learn the shapes of artificial characters placed on plain backgrounds; our semi-synthetic samples (Fig. 3.3c) are often placed on cluttered backgrounds and degraded due to sub-sampling/up-sampling, and thus represent an ideal point of connection between synthetic and natural data.

All those sets of heterogeneous samples are needed to reach an acceptable degree of accuracy for the ACF classifier, as shown by the Precision/Recall curves of Fig. 3.3d.

3.3.3 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; similarly to [9], a greedy Non-maximum Suppression (NMS) [10] is performed to discard overlapping regions.

In details: (i) we discard all the regions having score lower than $\mu(\{s_0, \ldots, s_n\})$; (ii) resize the remaining ones to half of their original size to obtain a good separation between detected text regions (see the considerations of [19] and the code of [9]); (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 [11] $IoU > 0.5$ with the one currently selected.

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 (as in [12]). 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. Even though this fixed threshold operation discards many true-positive regions (see Fig. 3.5), losing text components in *tcm* is not an issue, as those components can be fully recovered thanks to the ability of MR-MSER to detect entire words of text, as explained in Sec. 3.3.5.

3.3.4 Textness Map and MR-MSER

The text confidence map *tcm* is used, together with MR-MSER [37], 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 ∇) 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 sub-sampling 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 discard nested MR-MSER and retain only the left ones. This significantly decreases the final number of extracted MR-MSER: on average we discard more than 2000 components from the $\approx$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*.

(a) source (b) scale 1.0 (c) scale 0.5 (d) scale 0.25 (e) scale 0.12

Figure 3.6: MSER extracted at different levels of the pyramid capture different details: at low scales (≤ 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).

3.3.5 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; as in [25], we formulate an algorithm that can be applied to different datasets without extensive tuning.

We propose a peak-based text grouping algorithm that is extremely fast (see Table 3.1) and requires almost no parameterization.

In details: (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 intensity 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

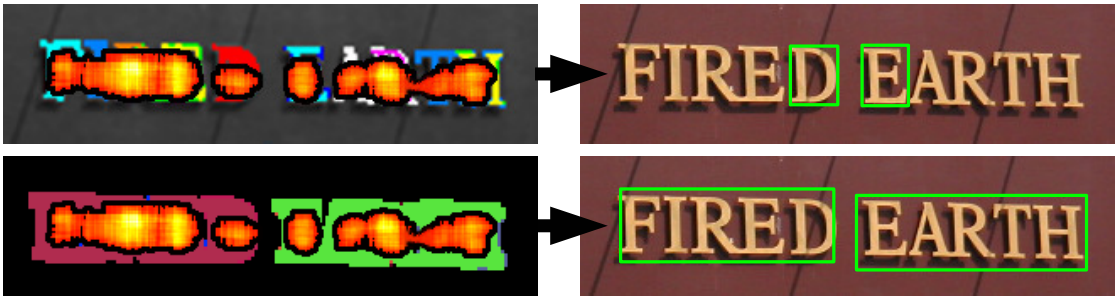


Figure 3.7: The text confidence map *tcm* is superimposed over MSER (top-left) and MR-MSER (bottom-left). Top-row: most characters are not identified when using MSER to reshape the connected components from *tcm*; Bottom-row: both words are correctly identified when using MR-MSER, as their intersection-over-union IoU with the connected components from *tcm* is greater than the given threshold.

size of the minimum bounding box that encloses all the 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. 3.3.3); (v) similarly to [25], neighbours connected components are merged into text lines using Mean-shift [72], connected components are clustered only on the basis of their centroid positions. The whole pipeline is summarized in Fig. 3.5.

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. From Fig. 3.6 we observe that MR-MSER extracted at low levels in the scale pyramid are 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 connected components.

As shown in Sec. 3.4.2 (see Table 3.2), MR-MSER outperform both state-of-the-art object proposal methods and MSER [33] at capturing entire words of text from natural images; our peak-based grouping algorithm exploits this talent to recover true-positive regions that are lost when removing false-negative regions, *e.g.* see Fig. 3.6: most of the components of the words *"Fired Earth"* are discarded when thresholding *tcm* (together with all the false-positive areas) however, since both words have been captured as single MR-MSER (see how *"Fired Earth"* entirely appears in *tex*), the proposed method successfully identifies their bounding boxes by using MR-MSER to reshape the partial bounding boxes identified from *tcm* to the boundaries of *"Fired"* and *"Earth"*.

Exploiting the word detection ability of MR-MSER to discard noise regions without worrying about losing true-positive areas is the key idea of our method.

Unlike most of the other methods for text localization, by clustering text components

Table 3.1: Implementation details. Times refer to a 640×480 image and ≈500 MR-MSER processed on a desktop Intel Core i5 with 12Gb RAM.

Task	Time (s)	Implementation
Gathering pos./neg. training data	103.40	Parallel
Training the classifier	185.58	Normal
Building the text confidence map	0.29	Parallel
Building the textness map	0.45	Parallel
Text line formulation	0.01	Parallel

on the basis of their centroid positions (ignoring scale, orientation, *etc.*), our algorithm often captures entire lines of text as single components. Even though some evaluation metrics penalizes our method for doing such thing (see Sec. 3.4.3), detecting text at line level is totally acceptable in real applications as the task of splitting lines into words can be carried out by more sophisticated text reading algorithms (*e.g.* PhotoOCR [22]).

3.3.6 Implementation Setup

Timings information for the proposed approach are given in Table 3.1: gathering positive/negative samples and training the classifier for ICDAR 2013 dataset require less than 5 minutes on a desktop machine (Intel Core i5, 12Gb RAM). On average, a natural 640×480 image can be fully processed in ≈0.75 seconds.

The computational complexity of the method can be reduced by decreasing the number of channels from which MR-MSER are extracted, by using a GPU implementation of the classifier (as in [61]), or by changing the implementation language.

The whole method has been developed using Matlab 2014a; training routines for the ACF classifier are implemented in C++ and used as external mex files.

3.4 Experimental Evaluation

In this section, we provide an experimental evaluation of the components described in Sec. 3.3: the performances of the ACF text region detector introduced in Sec. 3.3.1 are evaluated in Sec. 3.4.1; the capability of MR-MSER to detect words of text is analyzed in Sec. 3.4.2; the results achieved by the proposed approach for ICDAR 2003 and ICDAR 2013 text localization datasets are presented in Sec. 3.4.3 and compared with competing published and unpublished algorithms.

Table 3.2: Evaluation of MR-MSER for text word detection. 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 [38].

Image Channels	MR-MSER		MSER	
	chars	words	chars	words
∇	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

3.4.1 Classifier and Training Data

In Fig. 3.3d, we plot the Precision/Recall (PR) curves for multiple ACF classifiers trained using the same parameter configuration but different training samples.

PR curves have been obtained as in [19]: 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 2013.

This experiment shows how deeply the training data affects the performance of our sliding window classifier: poor results are obtained when training using just the samples from ICDAR 2013. Significantly better results are obtained when combining ICDAR's data with samples from Char74k; the best results are achieved when augmenting the positive training set with synthetic and semi-synthetic data generated as described in Sec. 3.3.2.

In every experiment we kept the training set balanced, meaning that the number of negative samples has always been equal to the number of positive samples.

The area-under-the-curve (AUC) for the PR curve for the classifier trained using natural, synthetic and semi-synthetic data is higher than the one of [19], this proves the effectiveness of Aggregated Channel Features and Fast Feature Pyramids for text localization from natural images (as expected from [59]).

3.4.2 Word Detection with MR-MSER

The proposed method heavily relies on the ability of MR-MSER to identify entire words of text from natural images (see Sec. 3.3.4 and Sec. 3.3.5). Similarly to MR-MSER and MSER, object proposal methods learn the concept of *object* and generate a set of windows that potentially contain that *object* in the processed image; for that reason

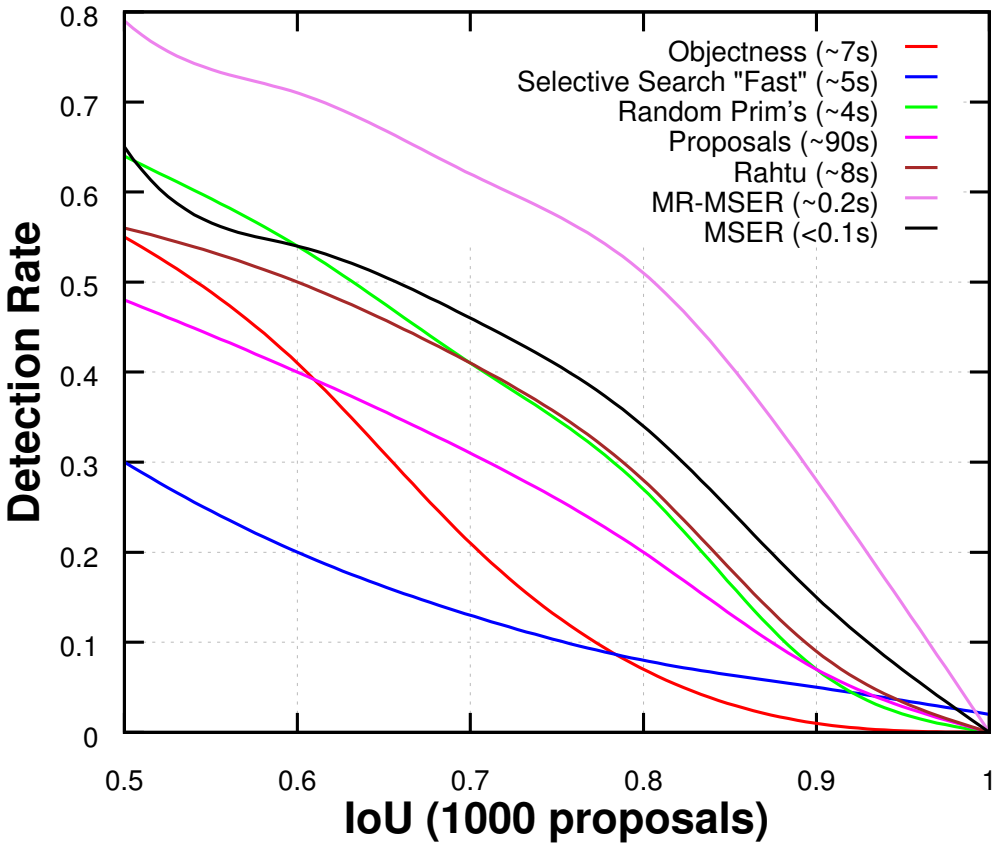


Figure 3.8: Text word detection accuracy evaluation and timings information for MR-MSER, MSER and object proposal methods on ICDAR 2003, while varying the intersection-over-union IoU coverage tolerance.

we decided to include the most popular object proposal algorithms in our experimental comparison.

In Fig. 3.8, MR-MSER are compared with object proposal methods [42, 44, 45, 46, 47] and MSER at detecting entire words of text from ICDAR 2003 images.

Results are measured as in [45]: for each algorithm, at most 1000 bounding boxes per image are selected from the ones initially extracted, the Detection Rate (DR, y-axis) is the percentage of ground-truth words *covered* by those bounding boxes. A ground-truth word is *covered* if it 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:

- Objectness [44]: 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;
- Selective Search [42]: evaluated in its *fast* variant, 1000 windows are uniformly sampled from the ones initially extracted;
- Prims [45]: a grid search is performed in $[0, 5]^3$ for color similarity, common border ratio and size, the bias is set to -3.00. The parameters providing the best results for 1000 unique windows and $IoU > 0.5$ are used for evaluation.
- Proposals [46]: evaluation is performed considering the bounding boxes surrounding the identified ranked segmentations proposals. The top 1000 windows are selected from the ones initially extracted.
- MR-MSER: extracted as described in Sec. 3.3.4, the bounding boxes surrounding each MR-MSER are considered for evaluation. On average, no more than 500 windows per image are generated.
- MSER [33]: extracted from RGB, HSI and ∇ channels. The bounding boxes surrounding 1000 unique MSER are uniformly sampled from the initial set.

MR-MSER prove their effectiveness as robust word detector from scene images by achieving higher detection accuracies throughout all the tolerance values; while all the other evaluated methods fail at $IoU \geq 0.6$.

3.4.3 Text Localization Results

In Tables 3.3a and 3.3b, the proposed text localization approach is evaluated on ICDAR 2003 and ICDAR 2013 datasets.

ICDAR 2003 [6] 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, Char74k, Synth and Semi-synth and 45000 negative samples from MSRC.

Bad training samples from ICDAR 2003 have been manually removed to avoid a degradation of performance.

Precision, Recall and F-measure are computed by looking for the best match between each detected bounding boxes and each ground-truth annotation [25, 29].

This evaluation metric penalizes approaches that detect text at line level, as only *one-to-one* (see [73]) matches are taken into account. As such, in order to obtain acceptable results, we disable the text component clustering step of our text line formulation algorithm (step (v) in Sec. 3.3.5); however, since MR-MSER often capture entire words as

Table 3.3: Text localization results for ICDAR 2003/2013 Challenge 2 Task 1.

(a) ICDAR'03. Evaluation metric: [6]

Method	P	R	F1
Li [25]	0.79	0.64	0.71
Kim [29]	0.78	0.65	0.71
Proposed	**0.71**	**0.74**	**0.70**
TD-Mixture [8]	0.69	0.66	0.67
Yi [74]	0.73	0.67	0.66
Epshtein [23]	0.73	0.60	0.66
Li	0.62	0.65	0.63
Chen	0.60	0.60	0.58
Neumann [32]	0.59	0.55	0.57
Zhang	0.67	0.46	0.55

(b) ICDAR'13. Evaluation metric: [7]

Method	P	R	F1
Proposed	**0.86**	**0.70**	**0.77**
Yin [31]	0.88	0.66	0.76
Neumann [75]	0.88	0.65	0.74
Bai [76]	0.79	0.68	0.73
Shi [30]	0.85	0.63	0.72
Shijian	0.75	0.69	0.72
Yang	0.70	0.65	0.67
Fabrizio [77]	0.74	0.53	0.62
Baseline	0.61	0.35	0.44
Inkam	0.31	0.35	0.33

single connected components, our method still generates a lot of *many-to-one* detections and therefore performs 1% worse than the best competing approach.

ICDAR 2013 [7] contains a total of 462 images: 229 for training and 233 for testing. The classifier is trained using 50000 positive samples from ICDAR2013, Char74k, Synth and Semi-synth and 50000 negative samples from MSRC.

Unlike ICDAR 2003, results are measured using the DetEval [73] software 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*). Unlike previous years, the concept of *don't care* zone is introduced to identify human unreadable characters that should not be taken into account to compute the final score.

Each method can be evaluated and compared with all the other published and unpublished works using the web page of the competition.

The proposed method outperforms all the competing approaches, full detection results for all the evaluated methods are available on ICDAR's web page.

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. 3.4.2, where multi-channel MR-MSER covers 78% of ICDAR's ground-truth words while MSER provides a coverage of 66%.

Note that results for ICDAR 2013 differ from those listed in [7] as in our table they have been updated using the latest correct values from the ICDAR 2013 website, and

(a) Positive (b) Negative

Figure 3.9: Examples of positive (a) and negative (b) text localization results from our method for ICDAR 2013 Challenge 2 Task 1 (green→true-positive, red→false-positive).

all the competing methods have been re-ranked according to their new F-score values. For references to all the algorithms see [7, 6].

Negative detection results are provided in Fig. 3.9, 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. 3.3.3: 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.

4

Text Spotting with Augmented MR-MSER Proposals and CNN

This chapter contains an overview of our article *"Augmented Text Character Proposals and Convolutional Neural Networks for Text Spotting from Scene Images"* [4] presented at the 3rd International Asian Conference on Pattern Recognition (ACPR2015) where it won the "best poster award".

4.1 Summary

In [4], unlike [64], we do not focus just on the task of text localization from natural images; instead we propose a novel method for end-to-end text localization and reading from scene images based on augmented Multi-resolution Maximally Stable Extremal Regions and Convolutional Neural Networks.

In this work we try to augment text character proposals to maximize their coverage rate over text elements in scene images, to obtain satisfying text detection and recognition rates without the need of using very deep architectures nor large amount of training data, unlike most of the other deep models proposed in literature for end-to-end text localization and recognition [43, 28].

Using simple and fast geometric transformations on multi-resolution proposals our system achieves good results for several challenging datasets while also being computationally efficient to train and test on a desktop computer.

A short introduction of our work is given in Sec. 4.2. The two main parts of the

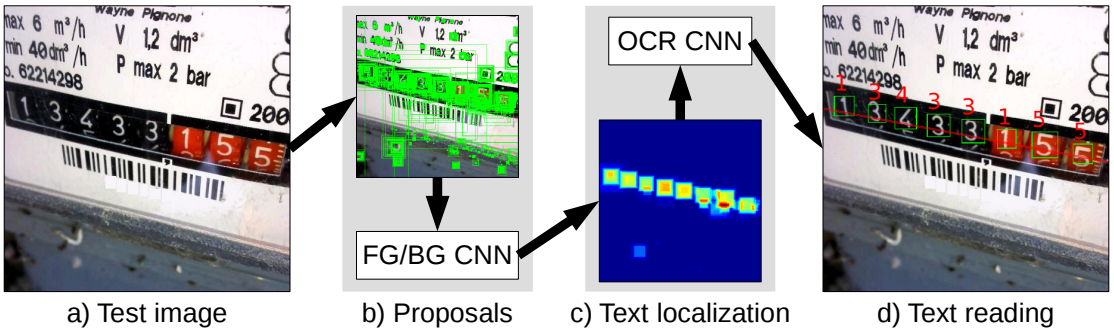


Figure 4.1: A visual overview of the proposed text spotting algorithm. Given a test image (a), augmented text character proposals (geometrically modified Multi-resolution Maximally Stable Extremal Regions) (b) are extracted and processed by a Convolutional Neural Network to build a text localization map in which potential areas of interest are highlighted (c). High intensity regions from the text confidence map are considered as potential regions of text and they are further processed to recognize text elements of interest (d).

pipeline for the proposed method are described in Sec. 4.3; an experimental evaluation of the method on several text spotting datasets is given in Sec. 4.4.

4.2 Introduction and Motivations

As discussed in Chap. 1, text localization and recognition (*text spotting*) from scene images and digital documents is an interesting task that finds applications in multiple commercial areas where automated systems can replace human workers in carrying out tedious repetitive data entry tasks.

In the last few years, researchers were able to obtain new state-of-the-art results for text spotting from scene images; however, recent state-of-the-art algorithms are often difficult to reproduce as they use very deep architectures [28] and/or large datasets (10 million or more manually tagged images) which sometimes are not publicly available due to copyright restrictions [78].

In our text spotting work, instead of focusing our attention on increasing either the deepness of the text localization and recognition classifiers, or the amount of labeled training data, we optimize the data that is fed to the proposed model by maximizing the detection recall of multi-resolution text character proposals extracted from scene images using simple geometric transformations.

Initially, we tackle the problem of reading analogic flow meters from natural images, showing that two slight variants of LeNet Convolutional Neural Network [53],

trained solely on augmented Multi-resolution Maximally Stable Extremal Regions (MR-MSER) [37], can reach nearly human detection accuracies and fast recognition times.

We then incrementally prove the generality of the proposed method by applying it to the task of license plate recognition [65, 66] and unconstrained text localization from scene images [7, 79, 43], obtaining state-of-the-art results for the first and competitive performances for the second.

In our experiments, for all the evaluated datasets, replacing augmented proposals with their respective non augmented versions leads to a dramatic reduction in terms of detection rates.

4.3 Algorithm

Text spotting is a complex task that requires an algorithm to detect and recognize all the natural and artificial text components appearing in the processed image.

In our method we approach this task using Convolutional Neural Networks. Most of the other works in literature perform end-to-end text localization and recognition in an integrated manner, using individual and extremely large deep architectures [43, 28] that require weeks or months to be trained on a professional server configuration equipped with expensive GPU (*e.g.* the network of [28] requires approximately one month to be trained using Google DistBelief [52]).

While these models work extremely well for the task they are trained, they are impossible to reproduce on a common desktop machines because the amount of video SDRAM required to load the network in the GPU exceeds the capacity of the GPU itself. Therefore, even if you have access to the trained model you cannot use it due to lack of memory resources.

In our method, instead of trying to reach the best possible text spotting detection rates, we focus on the development of a system that can be efficiently trained and used on a common desktop machine, and that can be easily extended to reach better performances on practical applications (*e.g.* unconstrained flow meter and license plate recognition).

To be able to reach acceptable results without using large Convolutional Neural Networks, we split the task of text spotting into two individual components using two indipendent networks, one trained for text localization (Sec. 4.3.1) and the other trained for text recognition (Sec. 4.3.2).

The first network is designed to build text confidence maps in which the intensity level associated with each pixel denotes the probability that the pixel belongs to foreground (text) or background (noise) The second network processes the information coming from those text confidence maps and associates a number/letter to each potential text element.

The whole pipeline (text localization and recognition) is visually summarized in

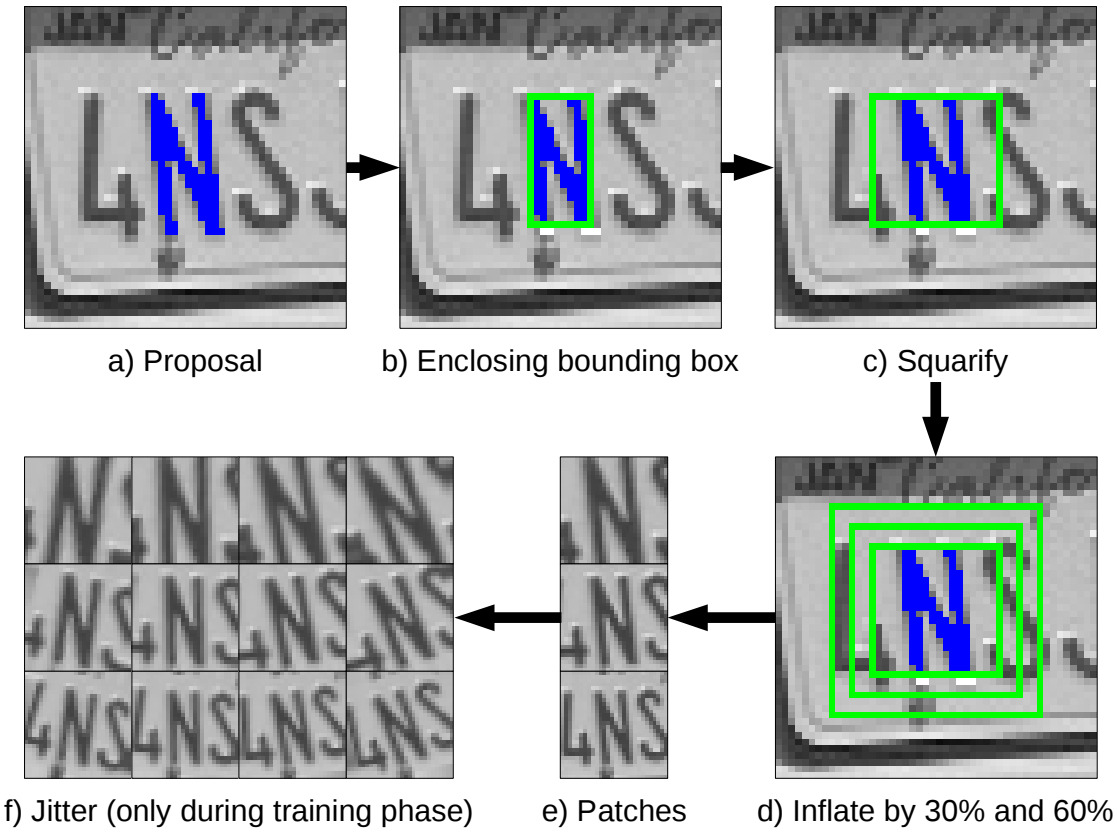


Figure 4.2: Text character proposal augmentation pipeline. Given a proposal and its bounding box (a,b): the bounding box is *squarified* without moving its center (c); two additionals bounding boxes are obtained by inflating the *squarified* bounding box by 30% and 60% of its area (d); resulting patches are randomly rotated within $[-\frac{\pi}{4}, \frac{\pi}{4}]$ to increase the robustness of the trained classifier (e,f).

Fig. 4.1 and deeply analyzed in the following sections.

4.3.1 Localization

The proposed text localization pipeline is visually summarized in Fig. 4.3.

Given a test image, Multi-resolution Maximally Stable Extremal Regions (MR-MSER) are computed as in [37]: Maximally Stable Extremal Regions (MSER) [33] proposals are extracted at each level of a scale pyramid, which has 1 octave per scale and a total of 3 scales.

Unlike [37, 64], no Gaussian smoothing is applied between octaves; Δ parameter, which regulates the amount of stability required for a connected component to be con-

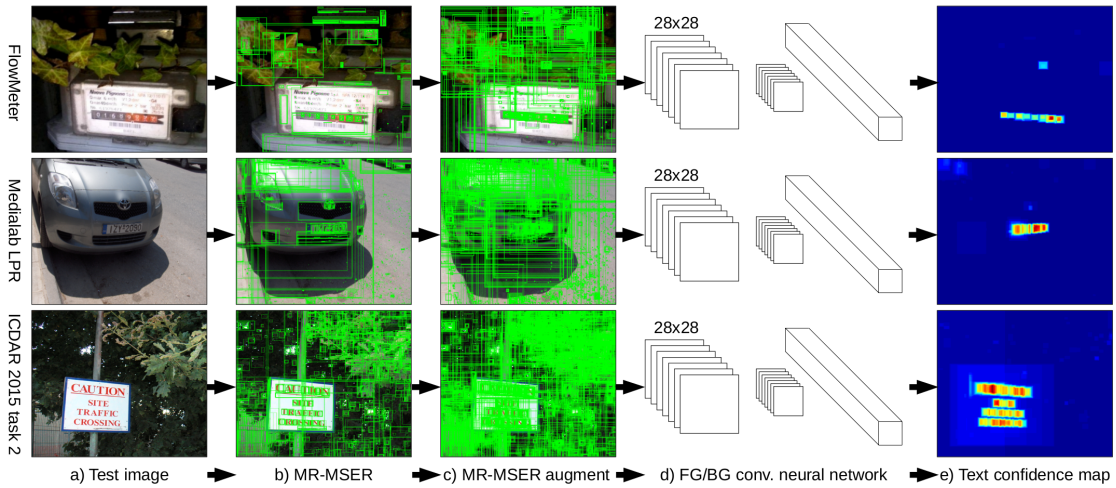


Figure 4.3: Text localization pipeline. MR-MSER text character are extracted from the given test images (a,b) are augmented using the augmentation pipeline of Fig. 4.2 (c) and processed by the text localization Convolutional Neural Network (d); foreground/background prediction values are stacked together to form text confidence maps (e).

sidered *stable*, is set to 3 to maximize the number of extracted proposals.

Using this setup, on average, roughly 8k MR-MSER proposals are extracted from a 640×480 rgb natural scene image.

The idea behind the use of MR-MSER is that unstable text regions in the original image may become stable at lower scales in the pyramid, where most image details are lost, colors are merged together [37], and even difficult text elements are captured as stable components (see Chap. 2 and the experimental evaluation provided in Chap. 3).

To increase detection recall of MR-MSER proposals over text regions in the processed images, we adopt the augmentation pipeline described in Fig. 4.2 : (i) the original proposal is *squarified* without moving its center, (ii) similar to [28], neighboring text characters and background noise are captured by inflating the *squarified* proposal by 30% and 60% of its area in every dimension, (iii) the *squarified* original proposal, together with its inflated variants, are resized to 28×28 pixel to fed as input to the Convolutional Neural Network.

Thanks to this augmentation routine, given a single MR-MSER proposal, a total of 3 augmented variants are generated; this provides us with roughly 24k image patches per image that need to be classified as either containing text characters of interest (foreground - FG) or noise (background - BG).

The task of classifying image patches as belonging to either FG or BG is approached using a slight variant of LeNet Convolutional Neural Network [53].

Table 4.1: Implementation details. Times refer to a 640×480 rgb image processed on a Intel Xeon E5-1620 at 3.5 GHz, NVIDIA GeForce GTX 980, and C++ Caffe Deep Learning library. The number of text character proposals first increases from 8k to 24k during the augmentation step described in Fig.4.2, and then drastically decreases when background components are filtered out after the text localization Convolutional Neural Network filtering step. The algorithm is roughly $10x$ faster than [64].

Task	Time (ms)	Comp.	# of prop.
MR-MSER (loc.)	14.4	CPU	8k
Prop. augment (loc.)	2.80	CPU	24k
FG/BG Convolutional Neural Network (loc.)	47.2	GPU	1k
OCR Convolutional Neural Network (read)	13.8	GPU	< 100

The proposed architecture has a total of three convolutional hidden layers with $[128, 256, 512]$ units each, and two fully connected layers containing 512 units. Max pooling with 2×2 window size is performed after each convolutional step. Kernel size and stride are fixed to respectively 3 and 1 for all the convolutional layers. The final classification is performed using Softmax which assigns the processed proposal a pair of values denoting the probability that the proposal belongs to FG and BG.

The network is trained using augmented MR-MSER proposals extracted from images from the given training dataset.

Positive samples are obtained by selecting augmented MR-MSER proposals having Intersection-over-union [11] $IoU > 0.5$ with at least one ground-truth text character annotation; an equal amount of negative training patches ($IoU = 0$ for every ground-truth text character annotation) are randomly selected from the given training dataset.

Note that, since the network is relatively small (10MB SDRAM), we can apply on-line jitter to the training patches while maintaining acceptable training speeds (≈ 1000 sample/s), and we can substantially speed up the training process by using a large batch size (256+ samples) during each training iteration.

Moreover, having a network that small allows us to use the proposed method even on cheap low-end GPU without worrying about the possibility of running out of GPU memory (most GPU avaible on the market have more than 256MB of SDRAM).

As shown in Fig. 4.2, each training patch is randomly rotated four different times within $[-\frac{\pi}{4}, \frac{\pi}{4}]$ radians, thus, starting from roughly 24k augmented patches we generate an average of 96k randomly rotated patches per training image.

As in [80] and in our previous work of Chap. 3, confidence values provided by the Convolutional Neural Network for each augmented MR-MSER proposal are stacked together to build a text confidence map in which high intensity regions denote potential

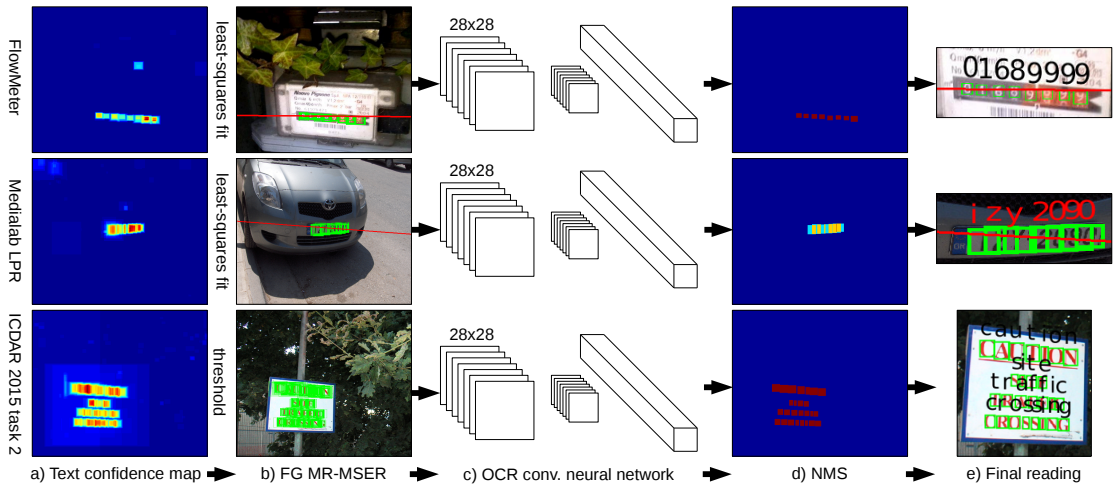


Figure 4.4: Text reading pipeline. Augmented text character proposals overlapping potential regions of text in text confidence maps (a,b), generated as in Fig. 4.3, are assigned a letter/number by the text reading Convolutional Neural Network (c). Non-maximum Suppression (NMS) [10] is performed over prediction scores generated by the network (d) to obtain the final readings (e).

text components of interest.

As shown in Fig. 4.3 and Table 4.1, the proposed localization pipeline works well for heterogeneous images, and requires on average 64.4 ms to be completed on a GTX 980 GPU, which is roughly $10x$ faster than our previous work on the localization.

4.3.2 Recognition

The proposed text reading pipeline is visually summarized in Fig. 4.4.

Given the normalized text confidence map (values in $[0, 1]$) produced by the previously described text localization step, we gather augmented MR-MSER proposals representing potential regions of text as follows: (i) each proposal is assigned a score computed as the average intensity of its pixels in the text confidence map, and (ii) proposals with score higher than σ are considered potential regions of text.

In our experiments we found that fixing $\sigma = 0.9$ gives an optimal balance between Precision and Recall on the evaluated datasets. However the value of σ should be changed if the goal is to obtain a more accurate system (higher Precision $\rightarrow$ higher σ), or to capture as many text elements as possible (higher Recall $\rightarrow$ lower σ).

Note that, since flow meter and license plate images contain single text lines of interest, to discard additional non-text proposals we compute the best fit line for the data using Weighted Linear Least-squares over proposal centers and scores, and remove

proposals that do not overlap that line. This routine cannot be used for ICDAR images as they may contain multiple lines of text (we only use threshold on ICDAR).

Non discarded proposals (roughly 1k per image) are then processed by a properly trained Convolutional Neural Network that performs OCR and assigns each of them a digit/letter together with a confidence value.

The network has the same architecture of the one used for text localization, and it is trained using the same data gathering and on-line jitter techniques.

Non-maximum Suppression (NMS) [10] is finally performed over proposal confidence values; NMS overlap threshold is set to 0.1 IoU to discard nested proposals generated by our augmentation technique.

Text reading take approximately 13.8 ms on a NVIDIA GeForce GTX 980 GPU. This is much faster than the text localization routine of Sec. 4.3.1 because, as showed in Table 4.1, the number of text character proposals that have to be processed by the text reading Convolutional Neural Network is roughly $1/24th$ of the number of proposals processed by the text localization Convolutional Neural Network.

4.4 Experimental Evaluation

The proposed method has been evaluated using the following three standard text spotting datasets: FlowMeter, Medialab LPR [65] and ICDAR 2015 task 2 [7].

In the next paragraphs the datasets are once again briefly introduced (a more complete description, together with some visual examples of images taken from those datasets, is provided in Chap. 2).

FlowMeter DB contains 6050 train and 168910 test scene images of gas flow meters. All the images were acquired using smart phones, and typically contain non-horizontal flow meters as well as difficult light conditions, lack of focus, motion blur, reflections, gravel on the digits, *etc.*

Medialab LPR contains 680 scene images of car license plates, obtained by merging all the collections from Medialab website (as in [65, 66]). [1] Similarly to competing methods, none of those images were used for training our model; instead, we used a total of 790 manually tagged training images from Zemris DB and UCSD Car LPR datasets. [2]

ICDAR 2015 Task 2 contains 229 train and 233 test scene images of focused text, it has been the reference dataset for text localization for the last decade due to its difficulty and large number of competing approaches. [3]

Evaluation results for FlowMeter, Medialab LPR and ICDAR 2015 Task 2 datasets are listed in Tables 4.2, 4.3 and 4.4 respectively.

[1] http://www.medialab.ntua.gr/research/LPRdatabase.html
[2] http://vision.ucsd.edu/belongie-grp/research/carRec/car_data.html
[3] http://rrc.cvc.uab.es/?ch=2&com=evaluation

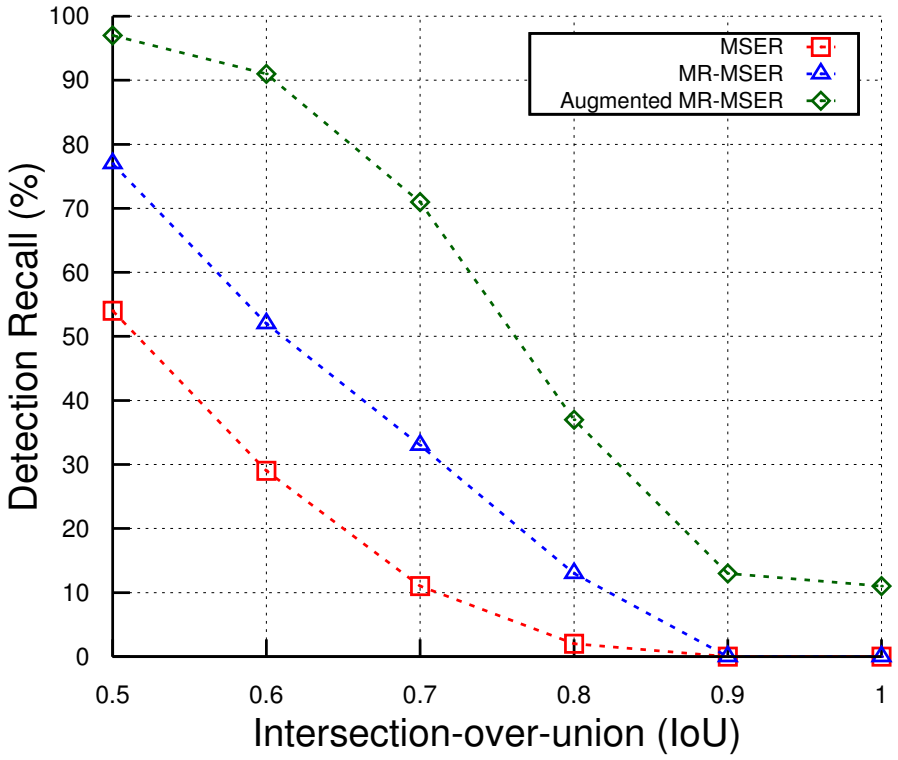


Figure 4.5: Text chatacter detection recall of MSER, MR-MSER and augmented MR-MSER proposals for FlowMeter dataset, while varying Intersection-over-union [11] IoU coverage tolerance as in [44, 42] and Chap. 3.

Results for Tables 4.2 and 4.3 are measured using Sequence Transcription Accuracy metric [28], namely the rate of test images for which the predicted sequence of numbers/letters matches their respective ground-truth data.

Recall (R), Precision (P) and Hmean for Table 4.4 are measured using DetEval evaluation tool [7] following the standard end-to-end ICDAR evaluation protocol.

The proposed method achieves nearly human performances for FlowMeter dataset, state-of-the-art results for Medialab LPR, and competitive results for ICDAR 2015 Task 2. Results on this latest dataset do not exceed the ones obtained by more sophisticated methods [43, 79] but it has to noted that, unlike most competing approaches (see website), our model has been trained solely on samples gathered from the original training set in an extremely small amount of time and using a cheap GPU.

Unsurprisingly, accuracies drop when not using augmented proposals; in fact, as shown in Fig. 4.5, augmented MR-MSER achieves on average 20% more detection recall on FlowMeter dataset for all the evaluated IoU values, compared with MSER [33] and MR-MSER [37].

Table 4.2: FlowMeter Database results. 168910 test and 6050 train images.

Method	Acc. (%)	Speed (img/s)
Human performance	95.1	0.08
SVM+HOG	67.4	2.10
Proposed	**93.6**	**12.8**
Proposed (no augment)	**83.1**	**12.9**

Table 4.3: Medialab LPR results. 790 test images from UCSD and Zemris datasets.

Method	Acc. (%)	Speed (img/s)
Human performance	97.2	0.25
Anagnostopoulos *et al.* [65] *	86.0	3.60
Zhu [66] **	87.3	9.80
Proposed	**90.2**	**10.0**
Proposed (no augment)	**83.3**	**10.4**

* Intel Pentium IV at 3.0 GHz with 512 MB RAM.

** Intel Core 2 Duo at 2.4 GHz with 512 MB RAM.

Table 4.4: ICDAR 2015 Challenge 2 Task 2 results. 233 test and 233 train images.

Method	R (%)	P (%)	Hmean (%)
StradVision [79]	80.2	90.9	85.2
VGGMaxNet_cmb [43]	77.3	92.2	84.1
ABBYY OCR SDK v10	35.1	61.0	44.5
Proposed	**67.0**	**83.2**	**74.1**
Proposed (no augment)	**47.9**	**82.4**	**60.5**

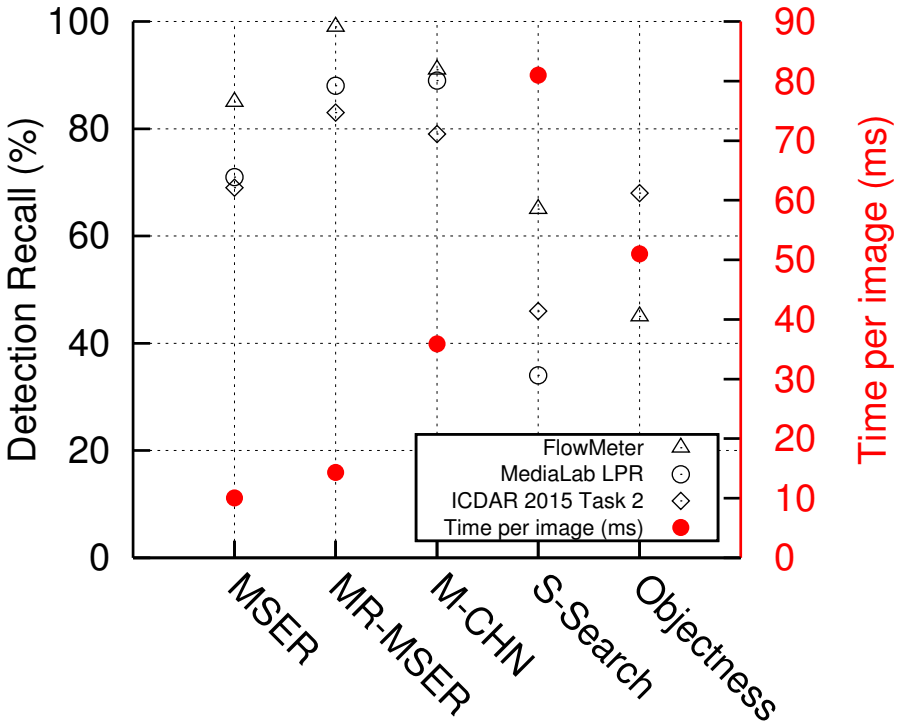


Figure 4.6: Text character detection recall evaluation and timings information for multiple augmented proposal algorithms [37, 38, 33]. In this experiment, intersection-over-union IoU is fixed and detection recall is computed at $IoU = 0.5$. MR-MSER [37] provide the best detection recall for all the evaluated datasets, with a small overhead in terms of computational complexity (roughly 3 ms more per image over simple MSER [33]).

As also shown in Fig. 4.6 and Table 4.1, augmented MR-MSER provides the best compromise between detection recall and computational complexity among the evaluated augmented proposal algorithms [37, 38, 33].

MR-MSER proves once again to be the best method for extracting text elements as connected components from natural scene images, requiring an overhead of roughly 3 ms to be computed for a 640×480 rgb image.

As in Chap. 3, Detection recall is measured as the percentage of ground-truth text character annotations *covered* by proposals; a text character is considered *covered* if there exists at least one proposal having $IoU > x$ with the ground-truth bounding box of that character; x varies on the horizontal axis in Fig. 4.5, and it is fixed to 0.5 in Fig. 4.6.

Using augmented MR-MSER proposals, our CNN provide text localization/reading predictions for each text character based both on the character patch (the original pro-

posal), and its surroundings (the inflated proposals); this is similar to processing the original image at a multi-resolution level and leads to more accurate text confidence maps with a small overhead in terms of computational completxity (see Table 4.1).

5

Conclusions, Future Directions and Practical Applications

5.1 Conclusions

Text localization and recognition from real-world images is a complex Computer Vision task that is being studied by many research laboratories and international companies for its importance and critical use in newly developed technologies, such as automated driving and automated indexing of information from visual data.

Unfortunately, to this day no method proposed in literature achieves text localization and recognition rates that are even remotely comparable to human observers' performances. For this very reason, competition among different text localization and recognition methods in this field is still very strong, especially on standard datasets like the ones from the International Conference of Document Analysis and Recognition (ICDAR) or the ones for real-world applications from Google, such as Google Street View House Numbers (SVHN) for house numbers localization and recognition from images harvested from Google Street View; Google Street View Text (SVT) for cropped lexicon-driven word recognition and full image lexicon-driven word detection and recognition from Google Street View.

As described in this thesis, throughout my Phd career two different approaches for text localization and text spotting from natural images were proposed by me and my research team [4, 64]. These methods achieved state-of-the-art performances on ICDAR and other challenging datasets at the time of their acceptance, and still provide excellent

localization and recognition rates when compared to other more recent and more complex works. In the following paragraphs, the pipelines and key insights that allow those methods to achieve those good recognition rates are briefly resumed and commented. In Sec. 5.2 we provide and motivate some ideas and advices on how these two methods can be modified and improved to reach even better results.

Like previously stated, text spotting from natural images finds many useful applications in real-world problems; as such, it is not surprising that the algorithm of Chap. 4 is currenly being used by different utility companies to gather information from visual data and replace tedious human activity. A brief description of the current industry application (automatic gas meter reading) of our method is given in Sec. 5.3.

In our work of Chap. 3 we exploit the latest advancements in rigid object recognition and Multi-resolution Maximally Stable Extremal Region (MR-MSER) to obtain state-of-the-art recognition rates for text localization from scene images. In this 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, our text localization system is computationally efficient to train and test, this enables its application to numerous problems in which execution and training times are critical factors.

In our work presented in Chap. 4 we expand our research interest to end-to-end text localization and recognition from natural scene images, proposing a novel and efficient deep-based method for text spotting. In this second work, our goal was to achieve good text detection and recognition rates in practical applications, paired with a low computational complexity. To this end, we introduced a novel fast geometric-based MR-MSER proposal augmentation technique which enhances detection recall of MR-MSER for text characters in natural scene images. Using small LeNet Convolutional Neural Network variants and augmented proposals, our system localizes and recognizes text characters of interest from 640×480 rgb images in roughly 78.2 ms on a desktop machine, can be fully trained in few hours (2-8, depending on the size of the processed training dataset), and achieves nearly human accuracies for several challenging text spotting datasets.

The two previously described methods leave room for many improvements, for example: (i) the forest of *stump* classifiers used in the work of Chap. 3 can be replaced with a faster and more accurate deep-architecture, this would most probably boost the text localization performance of the method and enable it to achieve state-of-the-art results for ICDAR 2015 Challenge 2 Task 1 text localization dataset; on the downside, it would also require a lot more of training samples than the ones currently used to train the forest (50k samples); (ii) the method of Chap. 4 can be extended by increasing the size

of the employed deep models (higher number of convolutional layers) and by extracting more training data so that it would be able to compete with other more accurate text spotting methods on ICDAR 2015 End to End Challenge 2 Task 4.

5.2 Future Directions

In my opinion, every possible improvement to the presented methods of Chap. 3 and Chap. 4 requires the introduction of deep architectures and the subsequent collection of more labelled training data. In fact, as shown in this thesis, deep-based approaches are currently the most effective and innovative way for reaching good results for text localization and recognition in natural scene images.

The positive side of these novel deep architectures is that their use allow text spotting methods to reach significantly higher detection rates than traditional shallow approaches. The negative side is that deep models require several million training samples to reach human-level performances and that all of the text spotting datasets currently available in literature contain at most thousands of labelled samples.

Given this situation, the most significative input for future developments of the presented works (and also for most of the competing deep-based text spotting approaches proposed in literature throughout these last years) would be to propose a method for automatically generating synthetic images of known text characters in known locations. In fact, as also stated by many experts in text localization and recognition fields, the best way to improve state-of-the-art results for deep-based text spotting algorithms consists in designing and developing a system for the automatic generation of synthetic train images that accurately simulate/mimic all the natural text elements and difficult conditions commonly found in real world natural scene images.

The way I see it, a possible solution for creating those images could be to exploit the advancements in computer graphics to artificially create highly realistic 3D environments like those found in modern simulation video games and then to capture from those 3D environments text elements from different point of views to simulate distortions, light variations, *etc.* as in real-world natural scene images. Such set of images would provide a virtually infinite set of positive samples that may be used to train deep models. The training accuracy of these models would probably keep increasing epoch after epoch because each individual sample is never seen twice during the different epochs, and the final trained classifier would most probably achieve human-level text spotting performances in practical applications.

The idea that more labelled training data leads to better recognition rates is not a myth as it has been proven multiple times in literature. For example, very recent deep-based works [22, 28] achieve recognition rates that are sometimes 20% higher than competing deep-based methods; one of the main reasons behind these outstanding results

lies in the large amount of labelled data used to train the deep models. Unfortunately for the research community, labelled data is precious, expensive to gather, and it is often publicly unavailable due to privacy issues. For example, the training labelled data of [22, 28] is not publicly available since it contains sensitive third party data. Most of that training data was gathered exploiting manual labelling from users and/or Amazon Mechanical Turk workers, both of these tagging techniques come at a high price.

Even though the lack of publicly available training data is the main issue, the extreme computational power required to train deep architectures is also a problem. Even supposing that a sufficient amount of labelled training data is made publicly available, some recently proposed deep architectures require an amount of computational power that is often achievable only by large IT companies. For example, the Convolutional Neural Networks of [28] require many weeks to be trained using Google DistBelief framework [52], and thus it is not feasible to re-train and/or tune that large model on smaller architectures in acceptable times. The same holds for many other deep architectures recently proposed in literature.

For these very reasons, if I had to begin another PhD course I would definitely work on the creation of a novel and publicly available dataset of synthetic samples that accurately mimic real world situations. I would then evaluate and compare the performances of deep methods trained on these synthetic samples against the results achieved by the same deep methods trained on natural data, expecting the first to be significantly higher than the latter due to the virtually infinite number of available samples and the large amount of heterogeneous difficult conditions.

5.3 Practical Applications

At the time of writing, the algorithm presented in Chap. 4 is being used by multiple companies to localize and recognize gas and water flow meters using NVIDIA CUDA GPU acceleration. In its current state, the algorithm processes million of images and achieves a processing speed rate of more than 10 images/second on a NVIDIA GTX 980, all while using less than 500MB of RAM.

Thanks to the large dimension of the test set extracted from the FlowMeter Database (Chap. 2.4.3, $\approx$ 170k images), the algorithm's recognition rates presented in [4] are similar to the ones it achieves in a real-world scenario: roughly 90% localization and reading success rate over more than 10 million processed gas meter images per year.

To achieve similar recognition rates on more difficult real-world applications using the same algorithm, larger training datasets need to be created. Based both on my experience and on the difficulty of the task, a training dataset of at least 10k tagged license plates may lead to human-level real-world detection rates for this text localization and recognition task. On the other hand, the problem of collecting a sufficient amount

of training data to reach human comparable detectiong rates for generic text localization and recognition from natural images is still open and it is currently being studied by many Computer Vision researchers.

Colophon

- This PhD thesis has been written using LaTeX.

- The LaTeX template for this thesis was made by Carullo Moreno.

- The algorithm of Chap. 3 was entirely developed using Matlab 2013b.

- The prototype for the algorithm of Chap. 4 was developed using Matlab 2014b; the final version of the same algorithm was written and deployed using C++.

- The following libraries have been used: Piotr Dollar's Computer Vision Matlab Toolbox for bounding box manipulation and AdaBoost classification [1]; Andrea Vedaldi's VLFeat for MSER computation [2]; Andrea Vedaldi's MatConvNet for CNN creation and manipulation in Matlab [3]; Berkeley Caffe for CNN creation and manipulation in C++ and Python [4]; NVIDIA Digits to provide an intuitive HTML front-end for CNN creation and training with Berkeley Caffe [5]; NVIDIA cuDNN for high performance neural network training using CUDA enabled GPU [6].

- For the work of Chap. 4, two NVIDIA GTX 980 were used: one donated by 7pixel and the other given to Artelab [7] by NVIDIA Corporation for research purposes.

- When not explicitly stated otherwise, all of the experiments of this thesis were executed on an Intel(R) Xeon(R) CPU E5-1620 v3 @ 3.50GHz with 64Gb of RAM and Linux Mint 17.1 Rebecca OS.

[1] http://vision.ucsd.edu/pdollar/toolbox/doc/
[2] http://www.vlfeat.org/
[3] http://www.vlfeat.org/matconvnet/
[4] http://caffe.berkeleyvision.org/
[5] https://developer.nvidia.com/digits
[6] https://developer.nvidia.com/cudnn
[7] http://artelab.dicom.uninsubria.it/

Bibliography

[1] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[2] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2010.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, pp. 1–42, 2015.

[4] A. Zamberletti, I. Gallo, and L. Noce, "Augmented text character proposals and convolutional neural networks for text spotting from scene images," in *Proceedings of the IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015.

[5] S. Lee, M. S. Cho, K. Jung, , and J. H. Kim, "Scene text extraction with edge constraint and text collinearity link," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2010.

[6] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competition," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2003.

[7] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Bigorda, S. Mestre, J. Mas, D. Mota, J. Almaz, and L. Heras, "ICDAR 2013 robust reading competition," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2013.

[8] C. Yao, X. Bai, W. Liu, and Y. Ma, "Detecting texts of arbitrary orientations in natural images." in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[9] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.

[11] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge 2012 results."

[12] Y.-F. Pan, X. Hou, and C.-L. Liu, "Text localization in natural scene images based on conditional random field," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2009.

[13] J. Sochman and J. Matas, "Waldboost - learning for time constrained sequential detection," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[15] C. M. Bishop, *Neural Networks for Pattern Recognition.* Clarendon Press, 1995.

[16] W. Niblack, *An Introduction to Digital Image Processing.* Strandberg Publishing Company, 1985.

[17] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.

[18] F. Yin and C.-L. Liu, "Handwritten chinese text line segmentation by clustering with distance metric learning," *Pattern Recognition*, vol. 42, no. 12, pp. 3146–3157, 2009.

[19] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2011.

[20] K. Wang and S. Belongie, "Word spotting in the wild," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[21] A. Crimisi, *Microsoft Research Cambridge Object Recognition Image Database*, 2004.

[22] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[23] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform." in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[24] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2009.

[25] Y. Li, W. Jia, C. Shen, and A. van den Hengel, "Characterness: An indicator of text in the wild," *IEEE Transactions on Image Processing (IP)*, 2014.

[26] T. Wang, D. J. Wu, Adam, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2012.

[27] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced mser tree," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[28] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[29] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and non-text filtering," *IEEE Transactions on Image Processing (IP)*, vol. 22, no. 6, pp. 2296–2305, 2013.

[30] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, "Scene text detection using graph model built upon maximally stable extremal regions," *Pattern Recognition*, vol. 34, no. 2, pp. 107–116, 2013.

[31] X.-C. Yin, X. Yin, and K. Huang, "Robust text detection in natural scene images," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[32] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2011.

[33] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2002.

[34] Y. Li and H. Lu, "Scene text detection via stroke width." in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2012.

[35] W. Huangx, Z. Liny, J. Yangy, and J. Wangy, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[36] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, pp. 91–110, 2004.

[37] P.-E. Forssén and D. G. Lowe, "Shape descriptors for maximally stable extremal regions," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[38] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[39] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2011.

[40] L. Ding and A. Goshtasby, "On the canny edge detector," *Pattern Recognition*, vol. 34, pp. 721–725, 2001.

[41] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[42] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," in *Proceedings of International Journal of Computer Vision (IJCV)*, 2013.

[43] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

[44] B. Alexe and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 11, pp. 2189–2202, 2012.

[45] S. Manen, M. Guillaumin, and L. V. Gool, "Prime object proposals with randomized prim's algorithm," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[46] I. Endres and D. Hoiem, "Category-independent object proposals with diverse ranking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 36, no. 2, pp. 222–234, 2014.

[47] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating object segmentation proposals using global and local search," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[48] S. Karaoglu, J. C. van Gemert, and T. Gevers, "Con-text: Text detection using background connectivity for fine-grained object classification," in *Proceedings of the ACM Multimedia Conference (ACMMM)*, 2013.

[49] L. Li, W. Feng, L. Wan, and J. Zhang, "Maximum cohesive grid of superpixels for fast object localization," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[50] F. Meng, H. Li, K. N. Ngan, and L. Zeng, "Feature adaptive co-segmentation by complexity awareness," *IEEE Transactions on Image Processing (IP)*, vol. 22, no. 12, pp. 4809–4824, 2013.

[51] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *International Journal of Computer Vision and Pattern Recognition (PAMI)*, 2013.

[52] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2012.

[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2012.

[55] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "Hoggles: Visualizing object detection features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[56] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," 2015.

[57] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[58] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[59] M. Mathias, R. Timofte, R. Benenson, and L. V. Gool, "Traffic sign recognition: How far are we from the solution?" in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2013.

[60] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[61] R. Benenson, M. Mathias, R. Timofte, and L. V. Gool, "Pedestrian detection at 100 frames per second," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[62] R. Benenson, M. Mathias, T. Tuytelaars, and L. V. Gool, "Seeking the strongest rigid detector," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[63] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 competition on robust reading," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[64] A. Zamberletti, I. Gallo, and L. Noce, "Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2014.

[65] C. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate recognition algorithm for intelligent transportation system applications," *IEEE Transactions on Intelligent Systems*, vol. 7, no. 3, pp. 377–392, 2006.

[66] S. Zhu, "An end-to-end license plate localization and recognition system," 2015, RIT.

[67] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision (IJCV)*, vol. 77, no. 1, pp. 157–173, 2008.

[68] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2012.

[69] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *Proceedings of International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2009.

[70] R. Appeal, T. Fuchs, P. Dollár, and P. Perona, "Quickly boosting decision trees pruning underachieving features early," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

[71] M. Villamizar, J. Andrade-Cetto, A. Sanfeliu, and F. Moreno-Noguer, "Bootstrapping boosted random ferns for discriminative and efficient object classification," *Pattern Recognition*, vol. 45, no. 1, pp. 3141–3153, 2012.

[72] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 5, pp. 603–619, 2002.

[73] C. Wolf and J. M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 8, no. 4, pp. 280–296, 2006.

[74] C. Yi and Y. Tian, "Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification," *IEEE Transactions on Image Processing (IP)*, vol. 21, no. 9, pp. 4256–4268, 2012.

[75] L. Neumann and J. Matas, "On combining multiple segmentations in scene text recognition." in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2013.

[76] B. Bai, F. Yin, and C. L. Liu, "Scene text localization using gradient local correlation." in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2013.

[77] J. Fabrizio, B. Marcotegui, and M. Cord, "Text segmentation in natural scenes using toggle-mapping." in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2009.

[78] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[79] M. C. Sung, B. Jun, H. Cho, and D. Kim, "Scene text detection with robust character candidate extraction method," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[80] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.