Hypergraph Modeling for Saliency Detection and Beyond

Yao Li

School of Computer Science

The University of Adelaide

A thesis submitted for the degree of

Master of Engineering Science

20/12/2013

Contents

Abstract

Salient object detection aims to locate objects that capture human attention within images. Previous approaches often pose this as a problem of image contrast analysis. In this work, we model an image as a hypergraph that utilizes a set of hyperedges to capture the contextual properties of image pixels or regions. As a result, the problem of salient object detection becomes one of finding salient vertices and hyperedges in the hypergraph. The main advantage of hypergraph modeling is that it takes into account each pixel's (or region's) affinity with its neighborhood as well as its separation from image background. Furthermore, we propose an alternative approach based on center-versus-surround contextual contrast analysis, which performs salient object detection by optimizing a cost-sensitive support vector machine (SVM) objective function. Experimental results on four challenging datasets demonstrate the effectiveness of the proposed approaches against the state-of-the-art approaches to salient object detection.

In addition to a novel method for salient object detection, we tackle scene text detection, a challenging research problem in the both vision and document analysis community, from the saliency detection prospective. Motivated by the need to consider the widely varying forms of natural text, we propose a bottom-up approach to the problem which reflects the 'characterness' of an image region. In this sense our approach mirrors the move from saliency detection methods to measures of 'objectness'. In order to measure the characterness we develop three novel cues that are tailored for character detection, and a Bayesian method for their integration. Because text is made up of sets of characters, we then design a Markov random field (MRF) model so as to exploit the inherent dependencies between characters. We experimentally demonstrate the effectiveness of our characterness cues as well as the advantage of Bayesian multi-cue integration. The proposed text detector outperforms state-of-the-art methods on a few benchmark scene text detection datasets. We also show that our measurement of 'characterness' is superior than state-of-the-art saliency detection models when applied to the same task.

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signature **Date**

Acknowledgements

First I would like to thank Dr. Chunhua Shen, my principal supervisor during my master study years. His dedication to research inspired me a lot, and will definitely influence me throughout my research career in the coming years. Under his supervision, not only have I learnt specific knowledge in the field, but I also know how to think as a mature researcher, including the ability to find unsolved challenges and address them in novel ways.

I would also like to thank staffs and visitors in the Australian Center for Visual Technologies (ACVT). As a non-native English speaker, I made grammar errors in academic manuscripts now and then. I would like to thank my co-supervisor, Prof. Anton van den Hengel, for his time and effort in revising those manuscripts. I also want to thank him for providing living cost to me for a year. In addition, I would like to thank Dr. Xi Li, a research fellow in our research center. A paper accepted by the International Conference on Computer Vision this year indicates that our cooperation is very successful. The paper is also a cornerstone of this thesis. Dr. Wenjing Jia is a lecturer from the University of Technology Sydney, who visited our lab for several months. She helped me a lot when I arrived in Adelaide. I would like thank her for her selfless help and our cooperation on research topic of scene text detection. Other staffs in the ACVT, such as Prof. Ian Reid and Dr. Anthony Dick, have given me thoughtful suggestions on research. In a word, I am very lucky to be a member of the ACVT.

As a master by research student, I spent most time with PhD students. Guosheng Lin, Zhen Zhang, Josh Boys, Fayao Liu, Zhenhua Wang, Quoc Huy Tran, Trung Thanh Pham, Julio Zaragoza, Anh Tuan Ngo

all have helped me a lot on both research and life. Having friends with different culture backgrounds really enriches my life experience.

Finally, thanks go to the excellent support from staffs in the School of Computer Science at the University of Adelaide. Pru Carter, Julie Mayo, Sharyn Liersch and Jo Rogers are administrative staffs of our school. They have offered me enormous support within the school such as booking flight and accommodation for conferences, which enables me focusing my main effort on research. Thanks also go to our Head of School, David Suter, and our Postgraduate Coordinator Frank Neumann. I am really grateful for what they have done.

List of Figures

Chapter 1

Introduction

1.1 Saliency detection

Visual attention, or visual saliency, is fundamental to the human visual system, and alleviates the need to process the otherwise vast amounts of incoming visual data. As such it has been a well studied problem within multiple disciplines, including cognitive psychology, neurobiology, and computer vision. In the vision community, image saliency detection aims to effectively identify important and informative regions in images. Early approaches in this area focus mainly on predicting where humans look, and thus work only on human eye fixation data [3, 4, 5, 6, 7]. Recently, a large body of work concentrates on *salient object detection* [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21], whose goal is to discover the most salient and attention-grabbing object in an image. This has a wide range of applications such as image retargeting [22, 1], image classification [23], and image segmentation [24]. Because it is difficult to define saliency analytically, the performance of salient object detection is evaluated on datasets containing human-labeled bounding boxes or foreground masks. Salient object detection is typically accomplished by image contrast computation, either on a local or a global scale. In general, local salient object detection [21, 18, 14] estimates the saliency degree of an image region by computing the contrast against its local neighborhood. Various contrast measures have been proposed, including mutual information [25], incremental coding length [26], and center-versus-surround

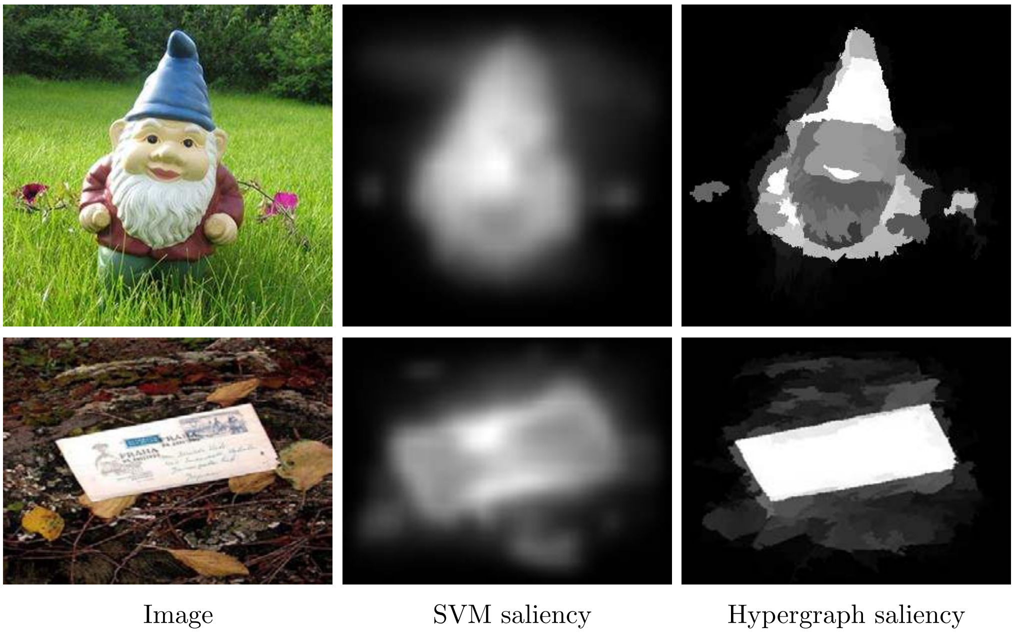


Figure 1.1: Illustration of our approaches to salient object detection.

feature discrepancy [11, 13, 14, 15, 18, 19, 20].

Global salient object detection approaches [9, 10, 12, 16, 17] estimate the saliency of a particular image region by measuring its uniqueness in the entire image. These approaches model uniqueness by exploiting the global statistical properties of the image, including frequency spectrum analysis [9], color-spatial distribution modeling [12], high-dimensional Gaussian filtering [16], low-rank matrix decomposition [17], and geodesic distance computation [10]. Therefore, the definition of object saliency depends on the choice of context. Global saliency defines the context as the entire image, whereas local saliency requires the definition of a local context.

1.2 Scene text detection

Human beings find the identification of text in an image almost effortless, and largely involuntarily. As a result, much important information is conveyed in this form, including navigation instructions (exit signs, and route information, for example), and warnings (danger signs *etc.*), amongst a host of others. Simulating such an ability for machine vision system has been an active topic in the vision and document analysis community. Scene text detection serves as an important preprocessing step for end-to-end scene text recognition which has manifested itself in various forms, including navigation, obstacle avoidance, and odometry to name a few. Although some breakthrough have been made, the accuracy of the state-of-the-art scene text detection algorithms still lag behind human performance on the same task.

Our basic motivation is the fact that *text attracts human attention*, even when amongst a cluttered background. This has been shown by a range of authors including Judd *et al.* [7] and Cerf *et al.* [27] who verified that humans tend to focus on text in natural scenes.

Previous work [28, 29, 30, 31] has also demonstrated that saliency detection models can be used in early stages of scene text detection. In [28], for example, a saliency map obtained from Itti *et al.* [3] was used to find regions of interest. Uchida *et al.* [31] showed that using both SURF and saliency features achieved superior character recognition performance over using SURF features alone. More recently, Shahab *et al.* [29] compared the performance of four different saliency detection models at scene text detection. Meng and Song [30] also adopted the saliency framework of [13] for scene text detection.

While the aforementioned approaches have demonstrated that saliency detection models facilitate scene text detection, they share a common inherent limitation, which is that they are distracted by other salient objects in the scene. The approach we propose here differs from these existing methods in that we propose a text-specific saliency detection model (*i.e.* a characterness model) and demonstrate its robustness when applied to scene text detection.

Measures of 'objectness' [15] have built upon the saliency detection in order to identify windows within an image that are likely to contain an object of interest. Applying an objectness measure in a sliding-window approach thus allows the identification of interesting objects, rather than regions. This approach has been shown to be very useful as a pre-processing step for a wide range of problems including occlusion boundary detection [32], semantic segmentation [33], and training object class detectors [34].

We propose here a similar approach to text detection, in that we seek to develop a method which is capable of identifying individual, bounded units of text, rather than areas with text-like characteristics. The unit in the case of text is the character, and much like the ‘object’, it has a particular set of characteristics, including a closed boundary. In contrast to the objects of [15], however, text is made up of a set of inter-related characters. Therefore, effective text detection should be able to compensate for, and exploit these dependencies between characters. The object detection method of [15] is similar to that proposed here in as much as it is based on a Bayesian framework combining a number of visual cues, including one which represents the boundary of the object, and one which measures the degree to which a putative object differs from the background.

In contrast to saliency detection algorithms which either attempt to identify pixels or rectangular image windows that attract the eye, our focus here is instead on identifying individual characters within non-rectangular regions. As characters represent the basic units of text, this renders our method applicable in a wider variety of circumstances than saliency-based paragraph detection, yet more specific. When integrating the three new characterness cues developed, instead of simple linear combination, we use a Bayesian approach to model the joint probability that a candidate region represents a character. The probability distribution of cues on both characters and non-characters are obtained from training samples. In order to model and exploit the inter-dependencies between characters we use the graph cuts [35] algorithm to carry out inference over an MRF designed for the purpose. Promising experimental results on benchmark datasets demonstrate that our characterness approach outperforms the state-of-the-art.

1.3 Overview of contributions

Our work involves two important topics in the vision community, *i.e.*, saliency detection and scene text detection.

Here, we propose two approaches to salient object detection based on hypergraph modeling and imbalanced max-margin learning. Our main contributions to saliency detection are as follows.

1. We introduce hypergraph modeling into the process of image saliency de-

tection for the first time. A hypergraph is a rich, structured image representation modeling pixels (or superpixels) by their contexts rather than their individual values. This additional structural information enables more accurate saliency measurement. The problem of saliency detection is naturally cast as that of detecting salient vertices and hyperedges in a hypergraph at multiple scales.

2. We formulate the center-surround contrast approach to saliency as a cost-sensitive max-margin classification problem. Consequently, the saliency degree of an image region is measured by its associated normalized SVM coding length.

We propose a novel scene text detection approach based saliency detection. Although previous work [28, 29, 30, 31] has demonstrated that existing saliency detection models can facilitate scene text detection, none of them has designed a saliency detection model tailored for scene text detection. We argue that adopting existing saliency detection models directly to scene text detection [28, 29, 30, 31] is inappropriate, as general saliency detection models are likely to be distracted by non-character objects in the scene that are also salient. In summary, contributions of our work on scene text detection comprise the following.

1. We propose a text detection model which reflects the ‘characterness’ (*i.e.* the probability of representing a character) of image regions. To our knowledge, we are the first to present a saliency detection model which measures the characterness of image regions. This characterness model is less likely to be distracted by other objects which are usually considered as salient in general saliency detection models.

2. We design an energy-minimization approach to character labeling, which encodes both individual characterness and pairwise similarity in a unified framework.

3. We evaluate ten state-of-the-art saliency detection models for the measurement of ‘characterness’. To the best of our knowledge, we are the first to evaluate state-of-the-art saliency detection models for reflecting ‘characterness’ in this large quantity.

1.4 Outline

This thesis will process as follow:

Chapter 2: Background. This chapter will cover some background knowledge on both saliency detection and scene text detection. As most recent literature on saliency detection, we will categorize state-of-the-art saliency detection approaches based on the the scope from which saliency is computed. We will give outlines of some representative saliency detection approaches from the large pool of literature. For scene text detection, we will divide existing scene text detection approaches into two groups, including texture-based and region-based approaches. When analysing the two different schemes, we review several algorithms whose results will be compared in the experiment.

Chapter 3: Contextual hypergraph modeling for salient object detection. In this chapter, we will first show that within a fixed context, a cost-sensitive SVM can accurately measure saliency by capturing center-surround contrast information. We will show that the use of a hypergraph captures more comprehensive contextual information, and therefore enhances the accuracy of salient object detection. In the experiment, we will show that the combination of the two proposed approaches yields significantly better result than the-state-of-arts using different evaluation criteria.

Chapter 4: Characterness: an indicator of text in the wild. In this chapter, we will present a scene text detection approach based on saliency detection. Specifically, we will first describe a characterness model, in which perceptually homogeneous regions will be extracted by a modified MSER-based region detector. Three novel characterness cues will be computed, each of which independently models the probability of the region forming a character. These cues will be fused in a Bayesian framework, where Naive Bayes is used to model the joint probability.

In order to consolidate the characterness responses we will design a character labeling method. An MRF, minimized by graph cuts [35], will be used to combine evidence from multiple per-patch characterness estimates into evidence for a single character or compact group of characters. Finally, verified characters will be grouped to readable text lines via a clustering scheme.

Two phases of experiments will be conducted separately in order to evaluate the characterness model and scene text detection approach as a whole. In the first phase, we will compare the proposed characterness model with ten state-of-the-art saliency detection algorithms on the characterness evaluation task, using evaluation criteria typically adopted in saliency detection. In the second phase, as in conventional scene text detection algorithms, we will use the bounding boxes of detected text lines in order to compare against state-of-the-art scene text detection approaches.

Chapter 2

Background

2.1 Saliency detection

The underlying hypothesis of existing saliency detection algorithms is the same: *the contrast between salient object and background is high.* Contrast can be computed via various features, such as intensity [14], edge density [15], orientation [14], and most commonly color [15, 21, 14, 13, 18, 11, 12, 20, 19, 16, 17, 10]. The measurement of contrast also varies, including discrete form of Kullback-Leibler divergence [14], intersection distance [11], χ^2 distance [15, 13, 20, 18], Euclidean distance [19]. As no prior knowledge about the size of salient objects is available, contrast is computed at multiple scales in some methods [19, 21, 18, 14, 13]. To make the final saliency map smoother, spatial information is also commonly adopted in the computation of contrast [19, 12, 11, 18, 16].

High level prior, which refers to the prior knowledge about where human may pay attention to when looking at an image, is often integrated in the saliency detection algorithms. Commonly used high level prior include center prior [17, 36] (objects near the image center are more attractive to people), face prior [17, 36] (people pay more attention to objects such as faces) and color prior [17, 36] (the warm colors such as red and yellow are more attractive to people). Based on the the scope of which the contrast is computed, The large amount of literature on saliency detection can be broadly classified into two classes, *i.e.*, local and global approaches.

2.1.1 Local approaches

Center-surround difference is the core of local saliency detection methods [3, 21, 18, 14], as they estimate saliency value of an image patch according to its contrast against its surrounding patches. The higher the difference of the center patch against the surrounding ones, the higher saliency value of the center patch is. As computing local contrast at one scale tends to only highlight boundaries rather than the whole object, local methods are always performed in the multi-scale manner. However, the computation of center-surround difference varies in different methods.

As a pioneer, Itti *et al.* [3] derived visual saliency as the center-surround difference of three features, including color, intensity and orientation on different scales. In their work, the center-surround difference was defined as the difference of filter responses between two scales in the image pyramid of a given feature. The final saliency map was constructed through the linearly combination of normalized feature maps. In [21], a rectangle window was divided into a rectangular inner window and the border, assuming that the inner window should be salient whereas the border belongs to the background. Similar to [3], the feature fusion idea was adopted in [14] where color, orientation and intensity features were used to compute the three saliency maps. They introduced discrete Kullback-Leibler Divergence from information theory as a measurement for center-surround difference. The work of [18] considered a superpixel was salient if it was distinguished from its immediate context, thus the saliency of a superpixel was computed as the sum of color contrast against all its spatial neighbors.

2.1.2 Global approaches

Global methods, e.g., [6, 9, 11, 12, 16, 17, 19, 10] take the entire image into account when estimating saliency of a particular patch. They estimate the saliency of a particular image region by measuring its uniqueness in the entire image. These approaches model uniqueness by exploiting the global statistical properties of the image, where globally rare features correspond to high saliency.

A typical global method was proposed by Cheng *et al.* [12]. They derived the saliency of a pixel as the saliency of the color of the pixel which was estimated by the sum of weighted color distance against all colors in the image. As the number of colors is huge (256^3 in RGB space), to make the computation feasible, they quantized the color into twelve bins in each channel. Furthermore, they proposed a region-based approach which incorporated spatial coherence. Some saliency detection methods reflect uniqueness from frequency domain [6, 9]. In [6], Xou and Zhang extracted the spectral residual of an image in spectral domain by analyzing the log-spectrum of an input image, then constructed the corresponding saliency map in the spatial domain. The work of Feng *et al.* [11] defined the saliency of a sliding window as the cost of composing the window using remaining parts of the image. An image was represented as a low-rank matrix plus sparse noises in [17] where the non-salient regions could be explained by the former while the latter component referred to salient regions. Perazzi *et al.* [16] proposed two measurements of saliency, *i.e.*, element uniqueness and distribution, which could be formulated within a single high-dimensional Gaussian filtering framework. In contrast with commonly-used center prior, Wei *et al.* [10] proposed boundary prior, assuming the image boundary is mostly background. Based on this assumption, the saliency of an image patch was defined as the length of its shortest patch to the image boundaries which was implemented by geodesic distance transform.

2.2 Scene text detection

Existing scene text detection approaches generally fall into one of two categories, namely, texture-based approaches and region-based approaches. While texture-based approaches are based on a top-down scheme, region-based approaches can be categorized into a bottom-up framework.

2.2.1 Texture-based approaches

Similar to many approaches in object detection, the main stages of texture-based approaches [37, 38, 39, 40, 41, 42] includes feature extraction, window classification and bounding box generation. In the feature extraction stage, as texture of texts is different from that of the background, conventional texture-based approaches extract some features from multi-scale sliding windows. Some widely-used features include Histograms of Gradients (HOGs), Local Binary Patterns (LBP), Gabor filters and wavelets. Rather than adopting above hand-engineered features, some works [41, 42] focused on unsupervised feature learning and deep learning. In the window classification stage, texture-based approaches dependent on trained classifiers, such as boosting [37, 38] and support vector machine (SVM) [41], to make the prediction of each sliding window. Finally, prediction scores in multi-scales are merged in some ways on the original scale to generate final bounding boxes in the bounding box generation stage.

In [38], the authors trained an adaboost classifier, which was a cascade with 4 strong classifiers containg 79 features. In [37], six types of features, including variance and expectation of X-Y ferivatives [38], local energy of Gabor filter and statistical texture measure of image histogram were employed to train a Modest AdaBoost classifier, where Classification and Regression Tree (CART) was used as the weak learner. As no prior knowledge of the size of text was available, sequential search via 16 scales sliding window was performed to handle variations in text size.

Instead of using hand-engineered features, recent years witnessed some texture-based approaches based on unsupervised feature learning and deep learning. To be more precise, to capture the texture properties of text on edges, Pan *et al.* [40] learnt a dictionary on text's edges via K-SVD and orthogonal matching pursuit (OMP) method. Given a patch from a test image, whether it belongs to text was measured by the sparsity of the feature vector learnt from the dictionary via OMP. Later on, Zhao *et al.* [39] improved the work of [40] by learning two discriminative dictionaries for text and background respectively. By comparing two reconstruction errors from two dictionaries, a particular patch from the test image was classified into the category with smaller reconstruction error. Similarly, instead of using heavily hand-engineered features, a variant of K-means clustering method was adopted in [41] to learn a dictionary from whitened 8-by-8 gray-scale patches. Then, the feature representation from the dictionary was used to train a linear SVM. More recently, Wang *et al.* [42] proposed an end-to-end scene text recognition system based on multilayer neural networks.

The biggest advantage of texture-based approaches is that their robustness to noise. Yet, there is something profoundly unsatisfying about texture-based approaches. The brute force nature of window classification is not particularly appealing. In other words, its computational complexity is proportional to the product of the number of scales.

2.2.2 Region-based approaches

In contrast with texture-based approaches which distinguish texts from background in sliding windows or image patches, a character segmentation step is adopted in region-based approaches initially [43, 44, 45, 46, 47, 2, 48, 49, 50, 51]. The character segmentation step extracts characters along with many non-text regions. After that, either simple geometric constraints [43, 44, 45, 46] or trained classifiers [49, 50, 51] are used to reject non-text regions. As a final step, remaining regions are clustered into lines through measuring the similarities between them. Two character segmentation techniques, Stroke Width Transform (SWT) [43] and Maximally Stable Extremal Region (MSER) [52], are commonly used in the state-of-the-art region-based scene text detection approaches.

The local image operator SWT is first introduced by [43], which has been successfully applied to later region-based approaches [49, 53]. The SWT method first finds the edges by Canny edge operator, then assigns each pixel with the most likely stroke width which is defined as the length of a line between two mostly perpendicular edge pixels. Connected components (CCs) are formed by grouping pixels with similar stroke width, followed by some simple heuristic rules to remove non-character regions. In [49], potential characters were extracted by the SWT initially. To reject non-characters two random forest classifiers were trained using two groups of features (component and chain level) respectively. In [2], the authors proposed an efficient way to extract stroke width of regions, which was based on skeletonization and distance transform. However, as SWT is largely dependent on the edge detection result, SWT cannot handle cases when text edges cannot be detected successfully, especially in images with much noise and low resolution.

As a character always has uniform color and some level of contrast against the background, many region-based approaches [47, 2, 54, 48, 51] use MSER as the character segmentation technique. In the work of Koo and Kim [51], candidate text regions were first extracted by MSER algorithm. Then, an Adaboost classifier was trained to determine the adjacency relationship and cluster regions by using their pairwise relations. Finally, non-characters were rejected by a trained multilayer perceptron classifier. The authors also found that using multichannel information improved the detection result. The text detection system of [54] consists of two stages: a hypothesis generation stage and a hypothesis verification stage. For hypothesis generation, to better handle image blur, they adopt MSER with a combination of judicious parameter selection and multi-scale analysis of MSER regions. For hypothesis verification, rather than using sophisticated classifiers or a large of number of features, they showed state-of-the-art result could be achieved by using a simple Gaussian model with only two well-motivated features. In contrast with most region-based approaches that compute features within extracted regions, Li *et al.* [48] introduced the concept of surrounding context after MSER detection. Surround context, in their definition, referred to color information in the surround background. Unary and pairwise surround context were combined to design an energy function which was minimized by the graph cut algorithm [35]. The optimal labels thus separated text and non-text regions.

In addition to SWT and MSER, some other character segmentation techniques in recent literature include local binarization [55, 56, 45] and color clustering [46, 50]. In [56], a text region detector first searched over different layers of the image pyramid. After projecting the text confidence and scale information back to the original image, candidate text regions were generated by scale-adaptive local binarization. Then, non-text regions were rejected in the Conditional Random Field (CRF) framework which incorporated unary region properties and binary contextual region relationships. Yi and Tian[50] introduced a boundary clustering technique for character segmentation, which was based on Gaussian mixture model (GMM) and EM algorithm to group the boundary pixels with bigram color uniformity on the border of text and attachment surface.

An advantage of region-based approach is that the result of character segmentation step can be sent to Optical Character Recognition (OCR) software for recognition directly, without the extra text extraction step. Another notable fact is that region-based approach are usually more computational efficient than their region-based counterparts, as the character segmentation step is scale insensitive. However, the biggest drawback is that state-of-the-art character segmentation techniques are usually sensitive to noisy and low-resolution images, which makes some texts missed in those images.

2.2.3 Scene text detection aided by saliency

In recent literature, some works [28, 31, 29, 30] verified that scene detection can be by aided by state-of-the-art saliency detection algorithms.

To our knowledge, the first tempt to combine scene text detection and visual attention model was made by Sun *et al.* [28]. In their work, candidate text regions were extracted by edge detection and connect component analysis. In the meantime, the classical visual attention model of Itti *et al.* [3] was adopted to compute saliency map. Regions of interest (ROIs) were generated by binarizing the saliency map, which were used as masks to filter out non-text regions.

Uchida *et al.* [31] proposed a keypoint-based approach towards scene text detection. They used SURF descriptors at key points to train an adaboost classifier. Their experimental results showed that using the feature vector which was a combination of SURF and saliency map outperformed using SURF alone.

More recently, Shahab *et al.* [29] evaluated the performance of four saliency detection models when applied to scene text detection. Not surprisingly, some models outperformed others on this task. Their conclusion was that saliency detection models can be used in the scene text detection.

Meng and Song [30] adopted the saliency detection framework of [13] to scene text detection. After obtaining the saliency map, they used Niblacks binarization algorithm to generate text regions. Non-text regions were rejected by the SVM classifier in the final step.

Chapter 3

Contextual hypergraph modeling for salient object detection

3.1 Cost-sensitive SVM saliency detection

As illustrated in [3, 21, 18, 14], saliency detection is typically posed as the problem of center-versus-surround contextual contrast analysis. To address this problem, we propose a saliency detection method based on imbalanced max-margin learning, which is capable of effectively discovering the local salient image regions that significantly differ from their surrounding image regions. In this case, the image is divided into overlapping rectangular windows which are tested for saliency. The context for each window is the windows that overlap it.

Before describing the method, we first introduce some notation used hereinafter. Let $\mathbf{x}_1$ denote the feature vector associated with a center image patch, and $\{\mathbf{x}_\ell\}_{\ell=2\ldots N}$ denote the feature vectors associated with the spatial overlapping surrounding patches of the center image patch. Using these patches, the proposed method explores their inter-class separability in a max-margin classification framework.

As shown in the top-right part of Fig. 3.1, the center image patch $\mathbf{x}_1$ is thought of as a positive sample while the surrounding patches $\{\mathbf{x}_\ell\}_{\ell=2\ldots N}$ are used as the negative samples. The saliency degree of $\mathbf{x}_1$ is determined by its inter-class separability from $\{\mathbf{x}_\ell\}_{\ell=2\ldots N}$. In other words, if $\mathbf{x}_1$ could be easily separated from

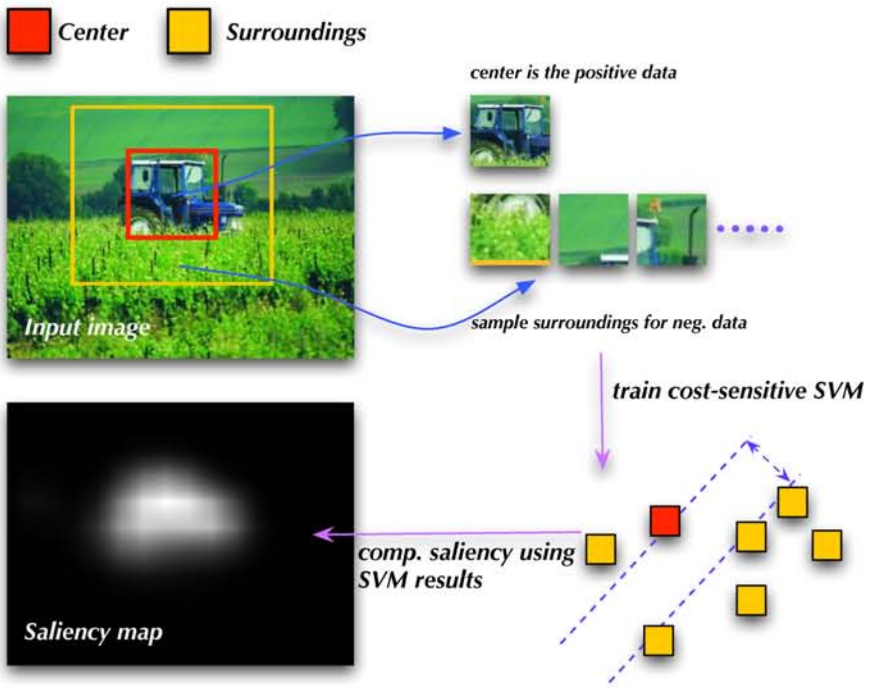

Center
Surroundings
center is the positive data
Input image
sample surroundings for neg. data
train cost-sensitive SVM
comp. saliency using SVM results
Saliency map

Figure 3.1: Illustration of cost-sensitive SVM for saliency detection. The saliency score is computed using based on the SVM classification results.

$\{\mathbf{x}_\ell\}_{\ell=2\ldots N}$, then it is deemed to be salient; otherwise, its saliency degree is low. This is a binary classification problem, which is associated with a cost-sensitive classification objective function [57]:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\epsilon}} \quad & J(\mathbf{w},b,\boldsymbol{\epsilon}) = \tfrac{1}{2}\|\mathbf{w}\|_2^2 + \tfrac{1}{2}C\textstyle\sum_{\ell=1}^{N} \nu_\ell \epsilon_\ell^2, \\ \text{s.t.} \quad & y_\ell = f(\mathbf{x}_\ell) + \epsilon_\ell, \end{aligned} \tag{3.1}$$

where $\|\cdot\|_2$ is the L_2 norm, $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ is the classifier to learn; $\boldsymbol{\epsilon}$ is the residual vector; C is the regularization parameter; and ν_ℓ is the corresponding weight of $\mathbf{x}_\ell$ such that $\nu_1 \gg \nu_\ell$ for $\ell = 2 \ldots N$. We set all the negative samples to have the same weight ν_ℓ, $\ell = 2 \ldots N$. According to the KKT condition, we have the following linear system:

$$\begin{bmatrix} 0 & \mathbf{1}_N^\top \\ \mathbf{1}_N & \Omega + V_C \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix}, \tag{3.2}$$

where $\mathbf{1}_N \in \mathcal{R}^N$ is the all-one column vector, $\mathbf{y} = (y_1, y_2, \ldots, y_N)^\top$ is the label vector, $\Omega = (\Omega_{ij})_{N\times N}$ is the kernel matrix $\Omega_{ij} = \mathbf{x}_i^\top \mathbf{x}_j$, and V_C is a diagonal matrix such that $V_C = \mathrm{diag}(\frac{1}{C\nu_1}, \frac{1}{C\nu_2}, \ldots, \frac{1}{C\nu_N})$. Based on the solution $(\boldsymbol{\alpha}^*, b^*)$ to the linear system (3.2), we have the weighted LS-SVM classifier $f(\mathbf{x}) = (\mathbf{w}^*)^\top \mathbf{x} + b^*$ with $\mathbf{w}^* = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)\boldsymbol{\alpha}^*$.

Using the weighted LS-SVM classifier $f(\mathbf{x})$, we define the saliency score as:

$$SSa(\mathbf{x}_1) = \frac{1}{N-1} \sum_{\ell=2}^{N} \frac{1 - \mathrm{sgn}(f(\mathbf{x}_\ell))}{2}, \tag{3.3}$$

where $\mathrm{sgn}(\cdot)$ is a sign function and the term $\sum_{\ell=2}^{N} \frac{1-\mathrm{sgn}(f(\mathbf{x}_\ell))}{2}$ counts the number of correctly classified surrounding samples. Loosely speaking, the saliency score $SSa(\mathbf{x}_1)$ can be viewed as a normalized SVM coding length (i.e., training accuracy for the surrounding samples), which characterizes the inter-class separability between $\mathbf{x}_1$ and its surroundings $\{\mathbf{x}_\ell\}_{\ell=2\ldots N}$. As shown in the bottom-left part of Fig. 3.1, the harder $\mathbf{x}_1$ is to separate from $\{\mathbf{x}_\ell\}_{\ell=2\ldots N}$, the smaller $SSa(\mathbf{x}_1)$ will be. In other words, the center patch looks similar to its surroundings. Conversely, the larger $SSa(\mathbf{x}_1)$ indicates the lower similarity between $\mathbf{x}_1$ and $\{\mathbf{x}_\ell\}_{\ell=2\ldots N}$, and hence a higher saliency degree. Note that, here the cost-sensitive LS-SVM is not the only choice. We can use other classifiers such as the exemplar SVM [58], where the standard hinge-loss SVM is used. We have used LS-SVM for its simplicity (it has a closed-form solution).

Example saliency maps derived from this measure are shown in Figs. 1.1 and 3.1. Although they accurately locate the salient object in each case, they also suffer from "fuzziness" or lack of precision around object boundaries and in locally homogeneous regions. This is mainly due to the center-surround local context that they are based on. In the next section, we describe an alternative approach based on segmentation based context that alleviates these problems.

3.2 Hypergraph modeling for saliency detection

To more effectively find salient object regions, we propose a hypergraph modeling based saliency detection method that forms contexts of superpixels to capture

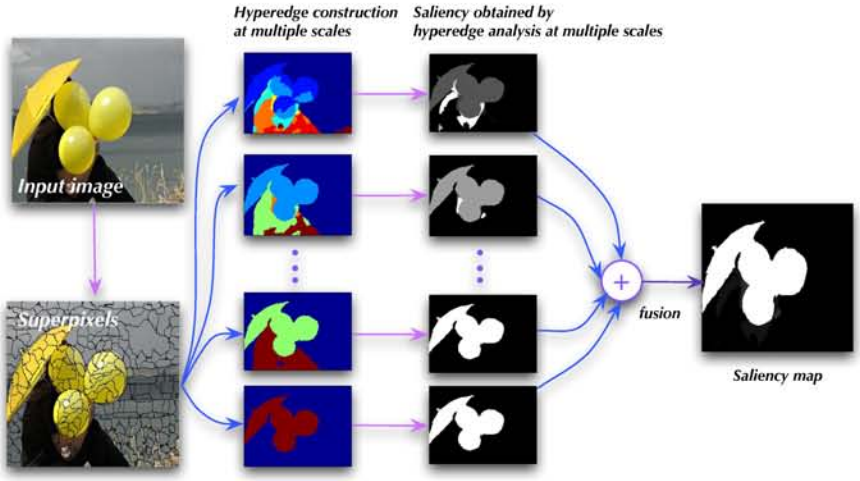

Hyperedge construction at multiple scales
Saliency obtained by hyperedge analysis at multiple scales
Input image
Superpixels
fusion
Saliency map

Figure 3.2: Illustration of hypergraph modeling for saliency detection using non-parametric clustering.

both internal consistency and external separation. Fig. 3.2 shows the high-level flowchart of the proposed method.

As illustrated in [59], a hypergraph is a graph comprising a set of vertices and hyperedges. In contrast to the pairwise edge in a standard graph, the hyperedge in a hypergraph is a high-order edge associated with a vertex clique linking more than two vertices. Effectively constructing such hyperedges is crucial for encoding the intrinsic contextual information on the vertices in the hypergraph.

3.2.1 Hypergraph modeling

In our method, an image I is modeled by a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i\}$ is the vertex set corresponding to the image superpixels and $\mathcal{E} = \{e_j\}$ is the hyperedge set comprising a family of subsets of $\mathcal{V}$ such that $\bigcup_{e \in \mathcal{E}} = \mathcal{V}$ [59]. As shown in Fig. 3.2, these hyperedges are constructed by multi-scale clustering, which groups the image superpixels into a set of superpixel cliques. Each clique corresponds to a collection of superpixels having some common visual properties, and works as a hyperedge of the hypergraph $\mathcal{G}$. The process of hyperedge construction implicitly encodes intrinsic affinity information on superpixels. Namely, if two superpixels have a higher co-occurrence frequency in the hyperedges, they tend to share more visual properties and have a higher visual similarity.

A hyperedge can also be viewed as a high-order context that enforces the contextual constraints on each superpixels in the hyperedge. As a result, the saliency of each superpixel, as measured by the hyperedges it belongs to, is not only determined by the superpixel itself but also influenced by its associated contexts. Due to such contextual constraints on each superpixel, we simply convert the original saliency detection problem to that of detecting salient vertices and hyperedges in the hypergraph $\mathcal{G}$. Mathematically, the hypergraph $\mathcal{G}$ is associated with a $|\mathcal{V}| \times |\mathcal{E}|$ incidence matrix $\mathbf{H} = (H(v_i, e_j))_{|\mathcal{V}|\times|\mathcal{E}|}$:

$$H(v_i, e_j) = \begin{cases} 1, & \text{if } v_i \in e_j, \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

The saliency value of any vertex v_i in $\mathcal{G}$ is defined as:

$$HSa(v_i) = \sum_{e \in \mathcal{E}} \Gamma(e) H(v_i, e), \tag{3.5}$$

where $\Gamma(e)$ encodes the saliency information on the hyperedge e. In essence, our hypergraph saliency measure (3.5) is a generalization of the standard pairwise saliency measure defined as:

$$PSa(v_i) = \sum_{v_j \in \mathcal{N}_{v_i}} d_{(v_i, v_j)} = \sum_{e \in \{(v_i, v_j) | j \neq i\}} \mathbb{I}_e d_e H(v_i, e), \tag{3.6}$$

where $\mathcal{N}_{v_i}$ stands for the neighborhood of v_i, $d_{(v_i, v_j)}$ measures the saliency degree of the pairwise edge (v_i, v_j), and $\mathbb{I}_e$ is the pairwise adjacency indicator (s.t. $\mathbb{I}_e = 1$ if $v_j \in \mathcal{N}_{v_i}$; otherwise, $\mathbb{I}_e = 0$). Instead of using simple pairwise edges, our hypergraph saliency measure takes advantage of the higher-order hyperedges (i.e., superpixel cliques) to effectively capture the intrinsic structural properties of the salient object, as shown in Fig. 3.3. To implement this approach, we need to address the following two key issues: 1) how to adaptively construct the hyperedge set $\mathcal{E}$; and 2) how to accurately measure the saliency degree $\Gamma(e)$ of each hyperedge.

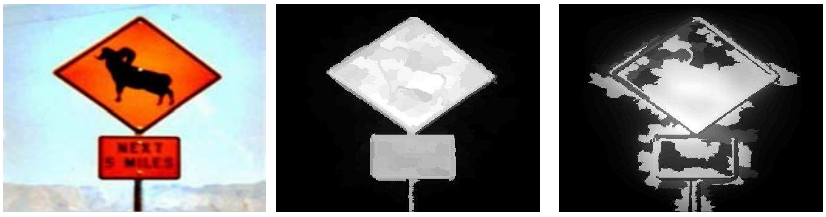


Image Hypergraph saliency Standard graph saliency

Figure 3.3: Illustration of salient object detection using two different types of graphs (i.e., hypergraph and standard pairwise graph). Clearly, our hypergraph saliency measure is able to accurately capture the intrinsic structural properties of the salient object.

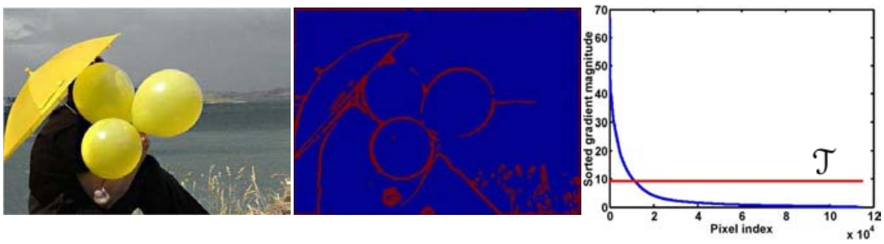


Figure 3.4: Illustration of the gradient magnitude information for hyperedge saliency evaluation. The left subfigure shows the original image, and the middle subfigure displays the gradient magnitude map I_g^* obtained by binarizing I_g using the adaptive threshold $\mathcal{T}$, as illustrated in the right subfigure.

3.2.2 Adaptive hyperedge construction

A hyperedge in the hypergraph $\mathcal{G}$ is actually a superpixel clique whose elements have some common visual properties. To capture the hierarchial visual saliency information, we construct a set of hyperedges by adaptively grouping the superpixels according to their visual similarities at multiple scales. In theory, this can be carried out in many ways using any number of established segmentation and clustering techniques. We adopt one such technique: non-parametric (mean shift) clustering.

Non-parametric clustering is typically associated with a kernel density estimator:

$$\widehat{f}_k(\mathbf{p}) = \frac{C_k}{Q|\mathbf{\Sigma}|^{\frac{1}{2}}} \sum_{i=1}^{Q} k(M^2(\mathbf{p}, \mathbf{p}_i, \mathbf{\Sigma})), \tag{3.7}$$

where $\mathbf{p}_i$ is a feature vector associated with the i-th superpixel (generated from image oversegmentation), $k(\cdot)$ is a kernel profile ($k(x) = \exp(-x/2)$ in our case), $\mathbf{\Sigma}$ is a symmetric positive definite bandwidth matrix (in the experiments, $\mathbf{\Sigma} = \gamma^2\mathbf{I}$ with γ being a scaling factor and $\mathbf{I}$ being an identity matrix), $M^2(\mathbf{p}, \mathbf{p}_i, \mathbf{\Sigma}) = (\mathbf{p} - \mathbf{p}_i)^\top \mathbf{\Sigma}^{-1}(\mathbf{p} - \mathbf{p}_i)$ stands for the Mahalanobis distance, and C_k is a normalization constant. Therefore, the superpixel cliques can be discovered by seeking the modes of $\widehat{f}_k(\mathbf{p})$. Mathematically, the mode-seeking problem can be converted to that of locating the zeros of the gradient $\nabla \widehat{f}_k(\mathbf{p}) = 0$, which leads to the following iterative procedure:

$$\mathbf{p}^{t+1} = \frac{\sum_{i=1}^{Q} g(M^2(\mathbf{p}^t, \mathbf{p}_i, \mathbf{\Sigma}))\mathbf{p}_i}{\sum_{i=1}^{Q} g(M^2(\mathbf{p}^t, \mathbf{p}_i, \mathbf{\Sigma}))}, \tag{3.8}$$

where $g(x) = -k'(x)$ and the superscript t indexes the iteration number. To accelerate the optimization process (3.8), we adopt a fast agglomerative mean-shift clustering method based on iterative query set compression [60].

Each mode is associated with a hyperedge, containing all the superpixels that converge to it after running the iterative procedure (3.8). The bandwidth matrix $\mathbf{\Sigma} = \gamma^2\mathbf{I}$ controls the scaling properties of the hyperedge. Consequently, using different values of γ for nonparametric clustering can generate the hyperedges at different scales, as shown in Fig. 3.2. By using different configurations of γ, we obtain a set of multi-scale hyperedges $\{e_i\}$ with e_i being the i-th hyperedge.

3.2.3 Hyperedge saliency evaluation

By construction, a hyperedge defines a group of pixels that is internally consistent. In addition, a salient hyperedge should have the following two properties: 1) it should be enclosed by strong image edges; and 2) its intersection with the image boundaries ought to be small [10, 18]. Therefore, we measure the saliency degree of a scale-specific hyperedge e by summing up the corresponding gradient magnitudes of the pixels (within a narrow band) along the boundary of the hyperedge. If the hyperedge touches the image boundaries, we decrease its saliency degree by a penalty factor.

More specifically, edge detection (using the Sobel operator in our case) is carried out for image I. Let I_x and I_y denote the x-axis and y-axis gradient

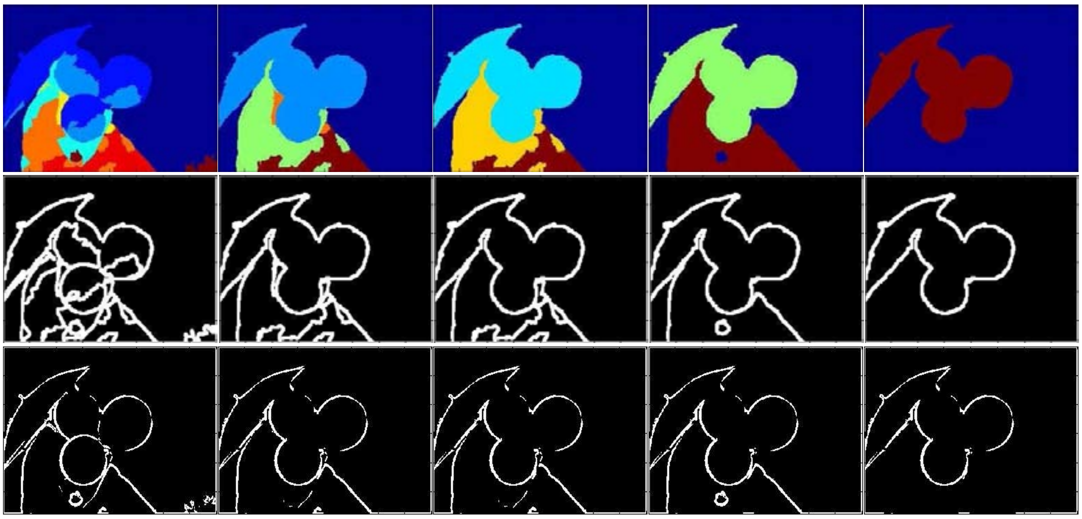


Figure 3.5: Illustration of M_g and $I_g^* \circ M_g$ for hyperedge saliency evaluation. The top row shows the multi-scale hyperedges; the middle row displays the scale-specific M_g that indicates the pixels (within a narrow band) along the boundary of the scale-specific hyperedge; and the bottom row exhibits the filtered gradient magnitude map $I_g^* \circ M_g$.

magnitude maps, respectively. Thus, the final gradient magnitude map I_g has the following entry: $I_g(m,n) = \sqrt{I_x^2(m,n) + I_y^2(m,n)}$. To obtain a robust gradient map, we introduce the following criterion: $I_g^*(m,n) = 1$ if $I_g(m,n) > \mathcal{T}$; otherwise, $I_g^*(m,n) = 0$, as shown in Fig. 3.4. Here, $\mathcal{T}$ is a threshold (picking out the top 10% of the I_g's elements in our case). As a result, the saliency value of the hyperedge e is computed as:

$$\Gamma(e) = \omega_e \left[\| I_g^* \circ M_g(e) \|_1 - \rho(e) \right]. \tag{3.9}$$

Here, ω_e is a scale-specific hyperedge weight (a larger scale leads to a larger weight), $\| \cdot \|_1$ is the 1-norm, $M_g(e)$ is a binary mask (illustrated in Fig. 3.5) indicating the pixels (within a narrow band) along the boundary of the hyperedge e, $\circ$ is the elementwise dot product operator, and $\rho(e)$ is a penalty factor that is equal to the number of the image boundary pixels shared by the hyperedge e. Based on Equ. (3.5), we obtain the hypergraph saliency measure $HSa(v_i)$ for any vertex v_i in the hypergraph $\mathcal{G}$.

3.3 Saliency fusion

After both SVM and hypergraph saliency detection, we obtain two saliency maps (i.e., SSa and HSa). Each element of these saliency maps is mapped into [0, 255] by linear normalization, leading to the normalized saliency maps (i.e., SSa^* and HSa^*). Based on such normalized maps, we define a saliency map fusion criterion as:

$$FSa = \lambda SSa^* + (1 - \lambda) HSa^*, \tag{3.10}$$

where λ is a trade-off control factor such that $0 \leq \lambda \leq 1$. Finally, the fused saliency map FSa is used for salient object detection.

3.4 Experiments

3.4.1 Experimental setup

Datasets As a subset of the MSRA dataset [13], MSRA-1000 [9] is a commonly used benchmark dataset for salient object detection. SOD [61] is composed of 300 challenging images. SED-100 is a subset of the SED dataset [62, 63], and consists of 100 images. Each image in SED-100 contains only one salient object. Imgsal-50 is a subset of the Imgsal dataset [64], and comprises 50 images with large salient objects for evaluation. Each image in the aforementioned datasets contains a human-labelled foreground mask used as ground truth for salient object detection.

Evaluation criterion For a given saliency map, we adopt four criteria to evaluate the quantitative performance of different approaches: precision-recall (PR) curves, F-measures, receiver operating characteristic (ROC) curves, and VOC overlap scores. Specifically, the PR curve is obtained by binarizing the saliency map using a number of thresholds ranging from 0 to 255, as in [9, 12, 17, 16]. As described in [9], F-measure is computed as $F = ((\beta^2 + 1)P \cdot R)/(\beta^2 P + R)$. Here, P and R are the precision and recall rates obtained by binarizing the saliency map using an adaptive threshold that is twice the overall mean saliency value [9]. $\beta^2 = 0.3$ is the same as that in [9]. Identical to [62], the ROC curve is generated from true positive rates and false positive rates obtained during the calculation of the corresponding PR curve. The VOC Overlap score [65] is defined as $\frac{|S \cap S'|}{|S \cup S'|}$. Here, S is the ground-truth foreground mask, and S' is the object segmentation mask obtained by binarizing the saliency map using the same adaptive threshold during the calculation of F-measure.

Implementation details In the experiments, cost-sensitive SVM saliency detection on an image is performed at different scales, each of which corresponds to a scale-specific image patch size for center-versus-surround contrast analysis. The final SVM saliency map is obtained by averaging the multi-scale saliency detection results. For computational efficiency, we first choose a fixed-sized image 8×8 patch and then resize the image using different downsampling rates to simulate the scale changes. In addition, each image patch is represented as a vectorized RGB feature vector. During the optimization process (3.1), the weight ν_1 for the center image patch is chosen as 0.5 while the weights ν_k (s.t. $k > 1$) for the surrounding image patches are set to 0.01, as suggested in [58]. Each superpixel $\mathbf{p}_i$ (referred to Equ. (3.8)) is first generated from image over-segmentation, and then represented by an 8-dimensional feature vector, which is obtained by averaging the corresponding color vectors of all the pixels in the superpixel. The color vector for each pixel contains four normalized color components $\mathbf{c} = (l, a, b, h)$ from the LAB and HSV color spaces, and thus has the form of $(\mathbf{c} \,|\, \mathbf{c}^{\frac{1}{3}})$ that is a concatenation of $\mathbf{c}$ and $\mathbf{c}^{\frac{1}{3}}$ (here $\mathbf{c}^{\frac{1}{3}}$ is an elementwise power transform [66]).

The scale-specific hyperedge weight ω_e (referred to in Equ. (3.9)) is determined by the scaling factor γ (mentioned in Sec. 3.2 for adaptive hyperedge construction). As for the hyperedge e, $\omega(e)$ is set to $2^\gamma/\mu$ with μ being a normalization constant such that $\mu = \sum_\gamma 2^\gamma$. The control factor λ in Equ. (3.10) is set to 0.15. We did not carefully tune the aforementioned parameters in the experiments. As shown in the supplementary file, our saliency detection approach is not sensitive to the choice of γ and λ. Note that the aforementioned parameters are fixed throughout all the experiments.

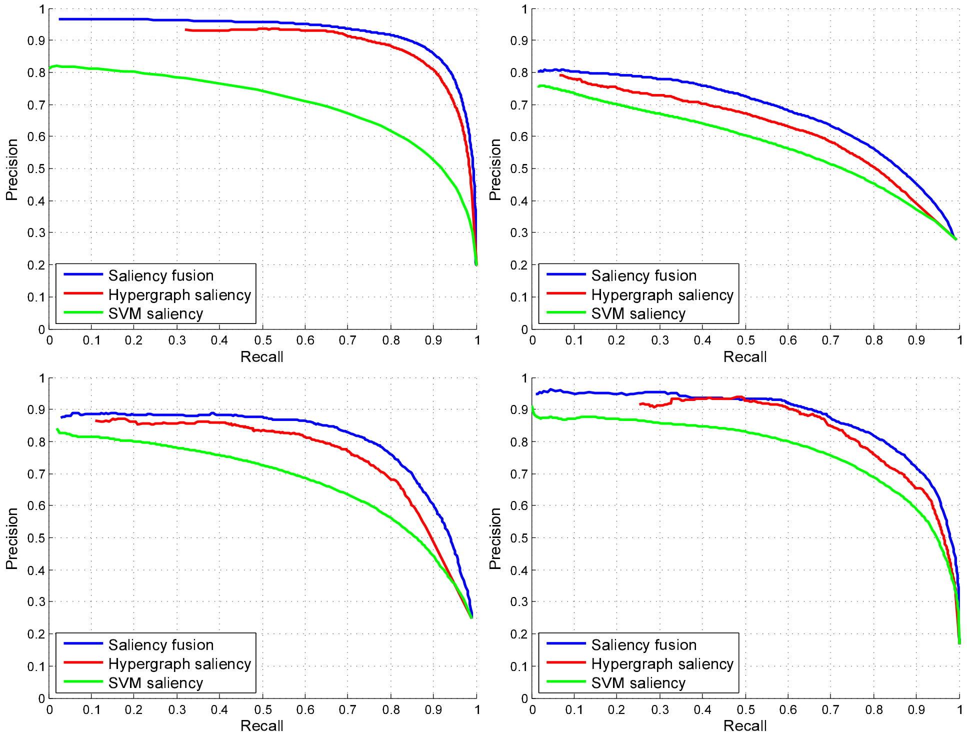
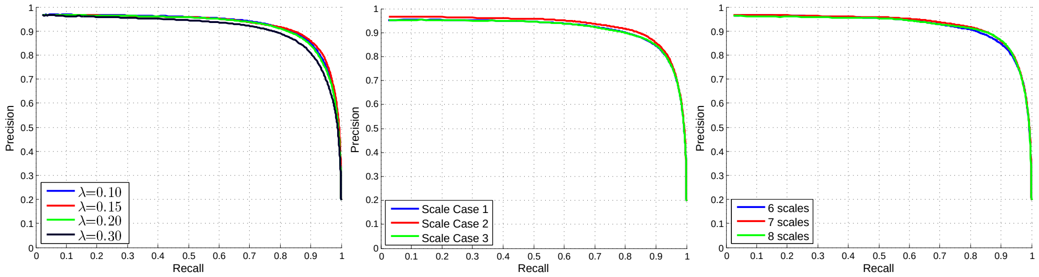


Figure 3.6: PR curves based on three different configurations: 1) using the SVM saliency approach only; and 2) using the hypergraph saliency approach only; 3) combining the SVM and hypergraph saliency approaches. Clearly, the saliency detection performance of using the third configuration outperform that of using the first and second configurations. From left to right: MSRA-1000, SOD, SED-100, and Imgsal-50.

3.4.2 Evaluation of individual approaches

Here, we evaluate the saliency detection performance of the proposed approaches based on three different configurations: 1) using the SVM saliency approach only; 2) using the hypergraph saliency approach only; and 3) combining the SVM and hypergraph saliency approaches. Fig. 3.6 shows their quantitative results of salient object detection in the aspect of PR curves. From Fig. 3.6, it is clearly seen that the saliency detection performance of only using the SVM saliency approach is significantly enhanced after combining the hypergraph saliency approach. The reason is that the hypergraph saliency approach captures both the internal consistency and strong boundary properties of salient objects. By incorporating the



(a) (b) (c)

Figure 3.7: Illustration of our saliency detection approach based on different parameter settings. (a) shows the PR curves of using different settings of λ; (b) displays the PR curves with different configurations of the scale space (determined by γ); and (c) exhibits the PR curves in different cases of scale numbers.

SVM saliency approach, the saliency detection results of only using the hypergraph saliency approach are further smoothed, leading to an improved saliency detection accuracy. Therefore, we use the best configuration (i.e., combination of SVM and hypergraph saliency) for performance evaluations in the following experiments. Fig. 3.9 shows some saliency maps of our SVM and hypergraph approaches from the four datasets.

3.4.3 Evaluation of different parameter settings

We evaluate the quantitative performance of our saliency detection approach using different parameter settings. Specifically, the quantitative evaluation task is carried out in the following three aspects: i) using different settings of the trade-off control factor λ in saliency fusion (see Eq. (10) of the paper); ii) using different configurations of the scale space for adaptive hyperedge construction (mentioned in Sec. 3 of the paper) with the same number of scales; and iii) using different numbers of scales during adaptive hyperedge construction.

1. As shown in Fig. 3.7 (a), we investigate the precision-recall (PR) performance of our approach by choosing different values of λ from the set $\{0.10, 0.15, 0.20, 0.30\}$. Note that $\lambda = 0.15$ is our default choice in the experiments of the paper. Fig From Fig. 3.7 (a), we see that the performance of our saliency detection approach is not sensitive to the choice of λ within a relatively wide range.

2. To demonstrate the quantitative performance difference using three different configurations of the scale space, we display the PR curves in Fig. 3.7 (b). These three configurations of γ are given as follows:

    - $\gamma \in \{0.10, 0.22, 0.30, 0.45, 0.52, 0.65, 0.78\}$,
    - $\gamma \in \{0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75\}$,
    - $\gamma \in \{0.12, 0.18, 0.31, 0.43, 0.51, 0.69, 0.79\}$.

    Note that the second configuration is used by our approach in the experiments. It is observed from Fig. 3.7 (b) that our saliency detection approach is not sensitive to the configuration of γ after moderate perturbation.

3. Fig. 3.7 (c) shows the quantitative PR curves using different numbers (i.e., 6, 7, and 8) of scales:

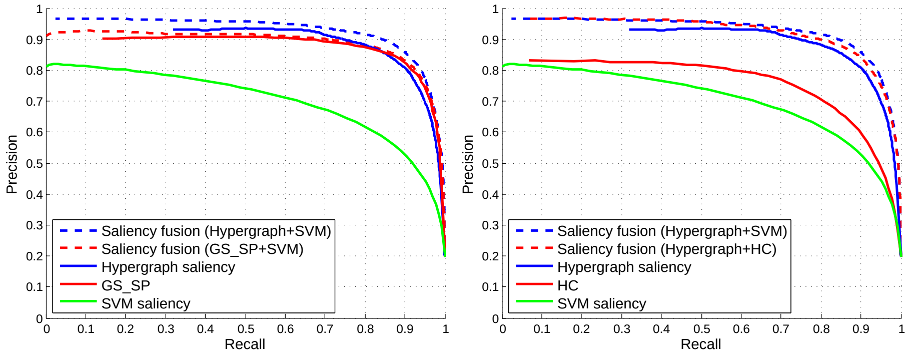
    - $\gamma \in \{0.15, 0.27, 0.39, 0.51, 0.63, 0.75\}$,
    - $\gamma \in \{0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75\}$,
    - $\gamma \in \{0.15, 0.24, 0.33, 0.42, 0.51, 0.60, 0.69, 0.75\}$.

    It is noted that our approach in the experiments takes the value of 7 by default. It is clear that the performance of our approach keeps relatively stable with respect to the choice of the scale number.

3.4.4 Evaluation of saliency fusion

In order to demonstrate the effectiveness of our saliency fusion (i.e., hypergraph+SVM), we evaluate the saliency detection performance of using the following two different configurations of saliency fusion: 1) varying the hypergraph saliency approach while keeping the SVM saliency approach fixed; and 2) changing the SVM saliency approach while fixing the hypergraph saliency approach.

For 1), we compare our hypergraph+SVM and GS_SP+SVM (that is a linear combination of our SVM saliency approach and the second best approach GS_SP [5]) on the MSRA-1000 dataset, as shown in Fig. 3.8 (a). From Fig. 3.8 (a), we see that both the hypergraph saliency and the GS_SP approaches on their own achieve a significantly higher performance than the SVM one. Meanwhile, the hypergraph saliency approach has a slightly higher performance than the GS_SP approach. In addition, the performance gain from combining the SVM saliency



(a) (b)

Figure 3.8: Evaluation of two different saliency fusion configurations on the MSRA-1000 dataset: 1) varying the hypergraph saliency approach while keeping the SVM saliency approach fixed; and 2) changing the SVM saliency approach while fixing the hypergraph saliency approach. (a) shows the PR curves of the saliency detection approaches associated with the first configuration while (b) displays the PR curves of the saliency detection approaches corresponding to the second configuration.

approach with the hypergraph saliency approach is greater than that obtained by combining the SVM saliency approach with the GS_SP approach.

For 2), we compare our hypergraph+SVM with hypergraph+HC (that is a linear combination of our hypergraph saliency approach and a contrast-based saliency approach [7]), as displayed in Fig. 3.8 (b). From Fig. 3.8 (b), we observe that again the hypergraph method outperforms both of the other standalone methods. In addition, the performance gain from adding SVM saliency to form hypergraph+SVM is greater than that from adding HC to form hypergraph+HC, even though the standalone performance of HC is higher than SVM.

In summary, our hypergraph+SVM achieves better performance than the other tested fusion configurations. Not only is the hypergraph method the best performing standalone method, but also its combination with the SVM method produces the most accurate overall result. Therefore, we conclude that our two approaches (i.e., SVM saliency and hypergraph saliency) are strongly complementary to each other.

3.4.5 Comparison with other approaches

In the experiments, we qualitatively and quantitatively compare the proposed approach with twelve state-of-the-art approaches, including GS_SP [10], LR [17], SF [16], CB [18], SVO [20], RC [12], HC [12], RA [21], FT [9], CA [19], ICL [26], and IT [3]. These approaches are implemented using their either publicly available source code or original saliency detection results from the authors.

Fig. 3.10 and Fig. 3.11 show the quantitative saliency detection performance of the proposed approach against the twelve competing approaches in the PR and ROC curves on the four datasets. From Fig. 3.10, we see that the proposed approach achieves the highest precision rate in most cases when the recall rate is fixed. Given a fixed false positive rate, the proposed approach obtains a higher true positive rate than the other approaches in most cases, as shown in the Fig. 3.11.

From Fig. 3.12, it is observed that the proposed approach achieves the best F-measure performance on the two popular benchmark datasets, that is, MSRA-1000 and SOD. On the SED-100 dataset, GS_SP and the proposed approach obtain the best results, and the F-measure of the proposed approach is slightly lower than GS_SP. On the Imgsal-50 dataset, the proposed approach is one of the two best approaches, and achieves a slightly lower F-measure than CB. In addition, Fig. 3.13 shows several salient object detection and segmentation (i.e., binarization using the adaptive threshold [9]) examples. It is seen from Fig. 3.13 that our approach obtain visually more feasible saliency detection results than the other competing approaches. From Fig. 3.13, we also observe that the proposed approach achieves the visually consistent segmentation results with ground truth. Furthermore, Tab. 3.4.6 shows the corresponding VOC overlap scores of all the thirteen approaches. It is seen from Tab. 3.4.6 that the proposed approach obtains the highest VOC overlap score with a low variance in most cases.

3.4.6 Application to image retargeting

The goal of image retargeting is to reduce image size while preserving important content. As shown in [1], saliency detection plays an important role in image retargeting. Following the work of [1], we directly replace its saliency detection component with ours while keeping the other components fixed. Fig. 3.14 shows some image retargeting examples of the two approaches (i.e., [1] and ours) on the image retargeting dataset from [1]. Clearly, our approach obtains more visually feasible results. This indicates that our approach is capable of effectively preserving the intrinsic structural information on salient objects during image retargeting.

Table 3.1: Quantitative performance of all the thirteen approaches in VOC overlap scores on the four datasets.

	MSRA-1000	SOD	SED-100	Imgsal-50
Ours	**0.77**±0.20	**0.40**±0.22	0.52±0.25	**0.69**±0.18
GS_SP [10]	0.75±0.22	0.38±0.20	**0.56**±0.27	0.65±0.21
LR [17]	0.63±0.25	0.29±0.19	0.41±0.27	0.64±0.18
SF [16]	0.67±0.24	0.27±0.20	0.47±0.27	0.59±0.22
CB [18]	0.72±0.24	0.31±0.25	0.52±0.32	0.64±0.19
SVO [20]	0.29±0.24	0.11±0.19	0.21±0.29	0.29±0.29
RC [12]	0.52±0.31	0.24±0.23	0.34±0.31	0.52±0.25
HC [12]	0.59±0.29	0.22±0.20	0.37±0.30	0.45±0.27
RA [21]	0.37±0.33	0.14±0.17	0.27±0.28	0.37±0.30
FT [9]	0.50±0.27	0.19±0.17	0.30±0.26	0.37±0.19
CA [19]	0.40±0.19	0.27±0.19	0.35±0.32	0.47±0.19
ICL [26]	0.33±0.19	0.22±0.17	0.34±0.22	0.30±0.21
IT [3]	0.17±0.12	0.14±0.11	0.16±0.14	0.19±0.10

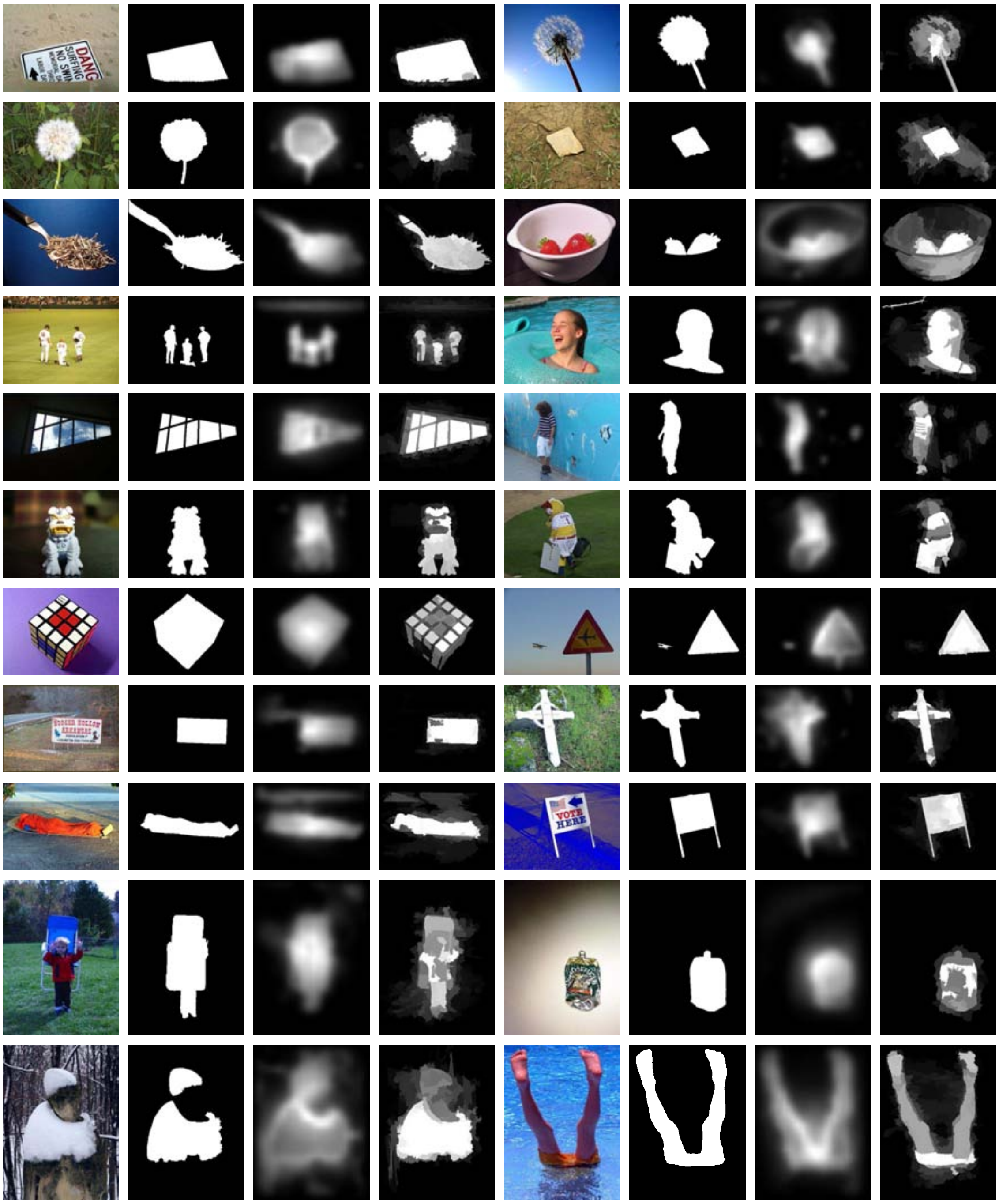


Figure 3.9: Saliency detection examples of our different approaches on the MSRA-1000 dataset. Clearly, the SVM saliency approach is able to locate the salient objects while the hypergraph saliency approach is capable of capturing the intrinsic structural information on the salient objects.

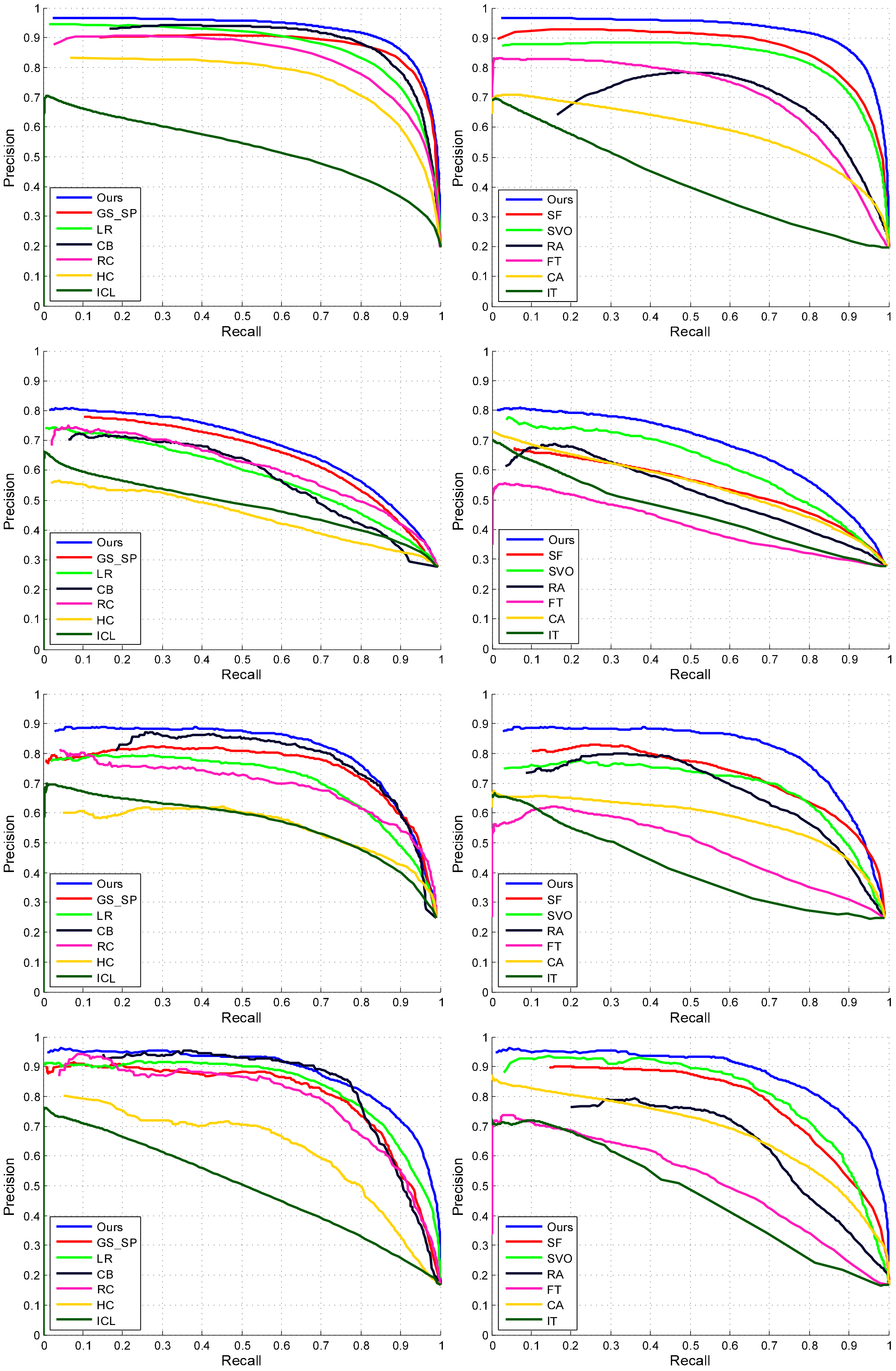


Figure 3.10: Quantitative PR curves of all the thirteen approaches on the four datasets. The rows from top to bottom correspond to MSRA-1000, SOD, SED-100, and Imgsal-50, respectively. Clearly, our approach achieve a better PR performance than the other competing approaches in most cases.

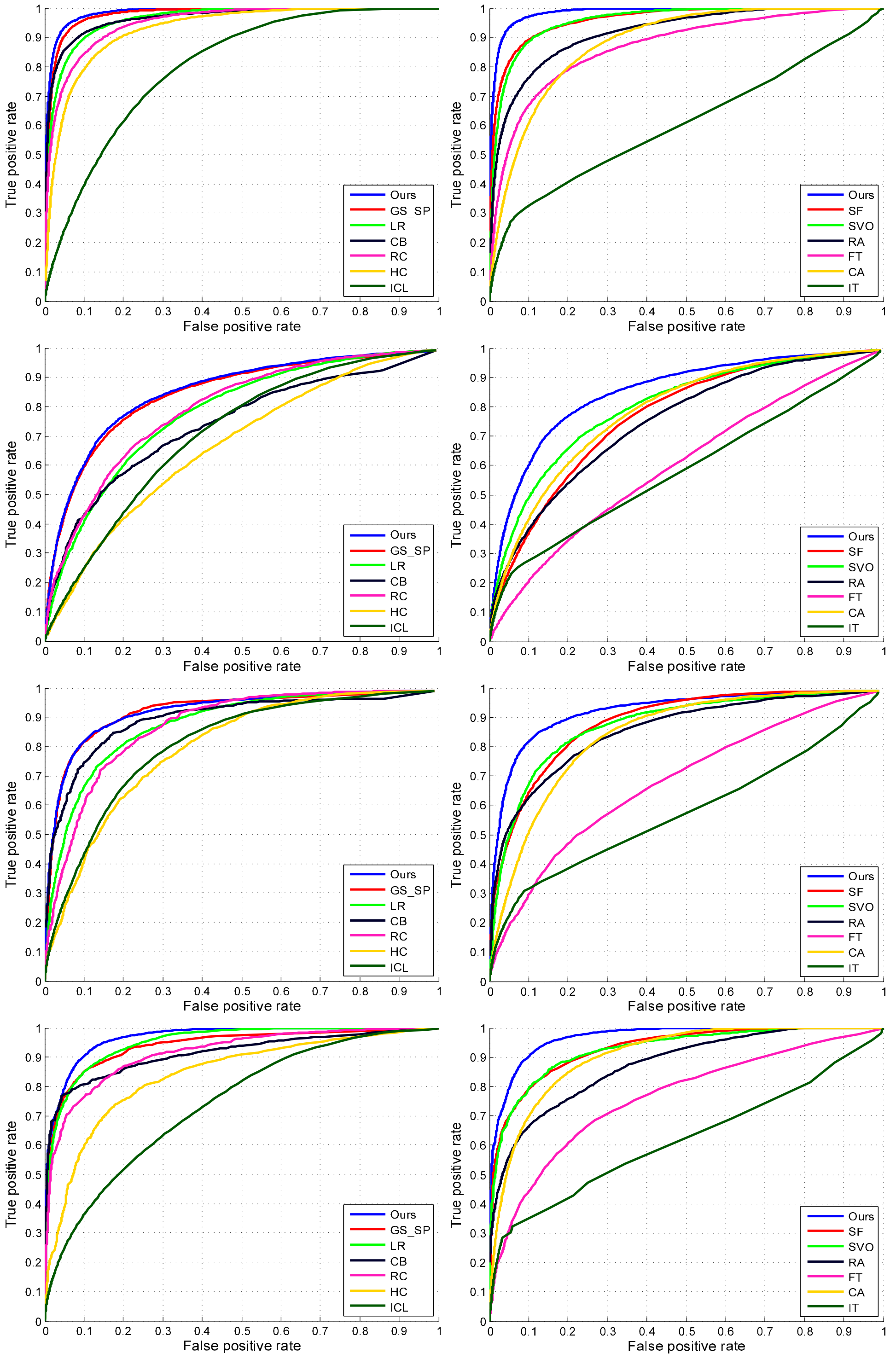


Figure 3.11: Quantitative ROC curves of all the thirteen approaches on the four datasets. The rows from top to bottom correspond to MSRA-1000, SOD, SED-100, and Imgsal-50, respectively. Clearly, our approach achieve a better ROC performance than the other competing approaches in most cases.

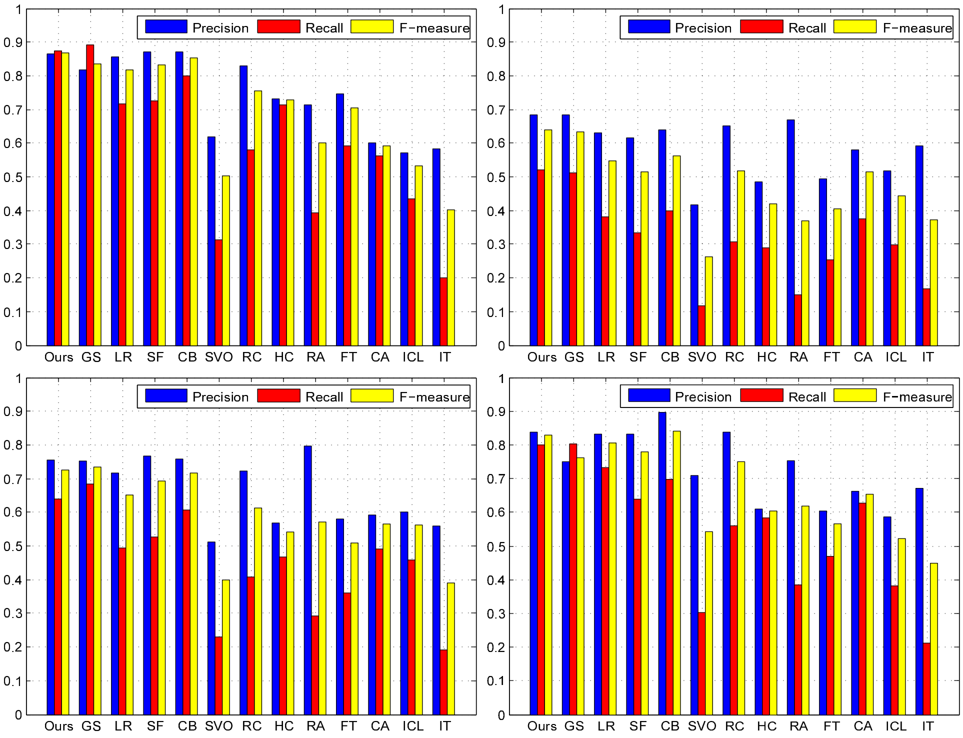


Figure 3.12: Quantitative F-measure performance of all the thirteen approaches on the four datasets. The columns from left to right correspond to MSRA-1000, SOD, SED-100, and Imgsal-50, respectively. Here, GS is a shorthand form of GS_SP. It is clear that our approach achieve a good F-measure performance on the four datasets.

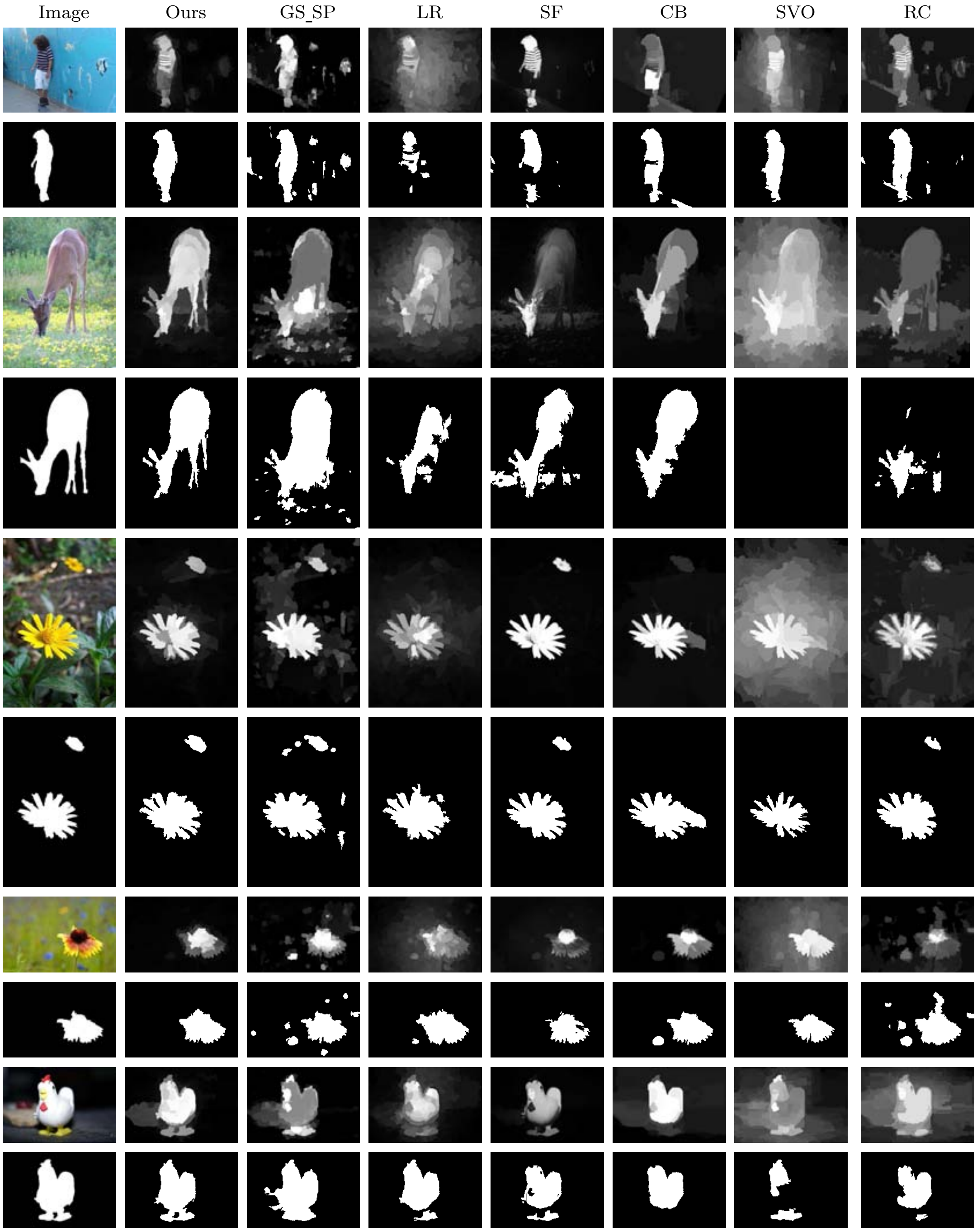


Figure 3.13: Salient object detection and segmentation examples on the MSRA-1000 dataset. For each example, the top row shows the input image and its corresponding saliency maps obtained by different approaches, and the bottom row displays the ground truth and the salient object segmentation results associated with the saliency maps. It is clear that our approach obtains the visually more consistent saliency detection and segmentation results than the other competing approaches.

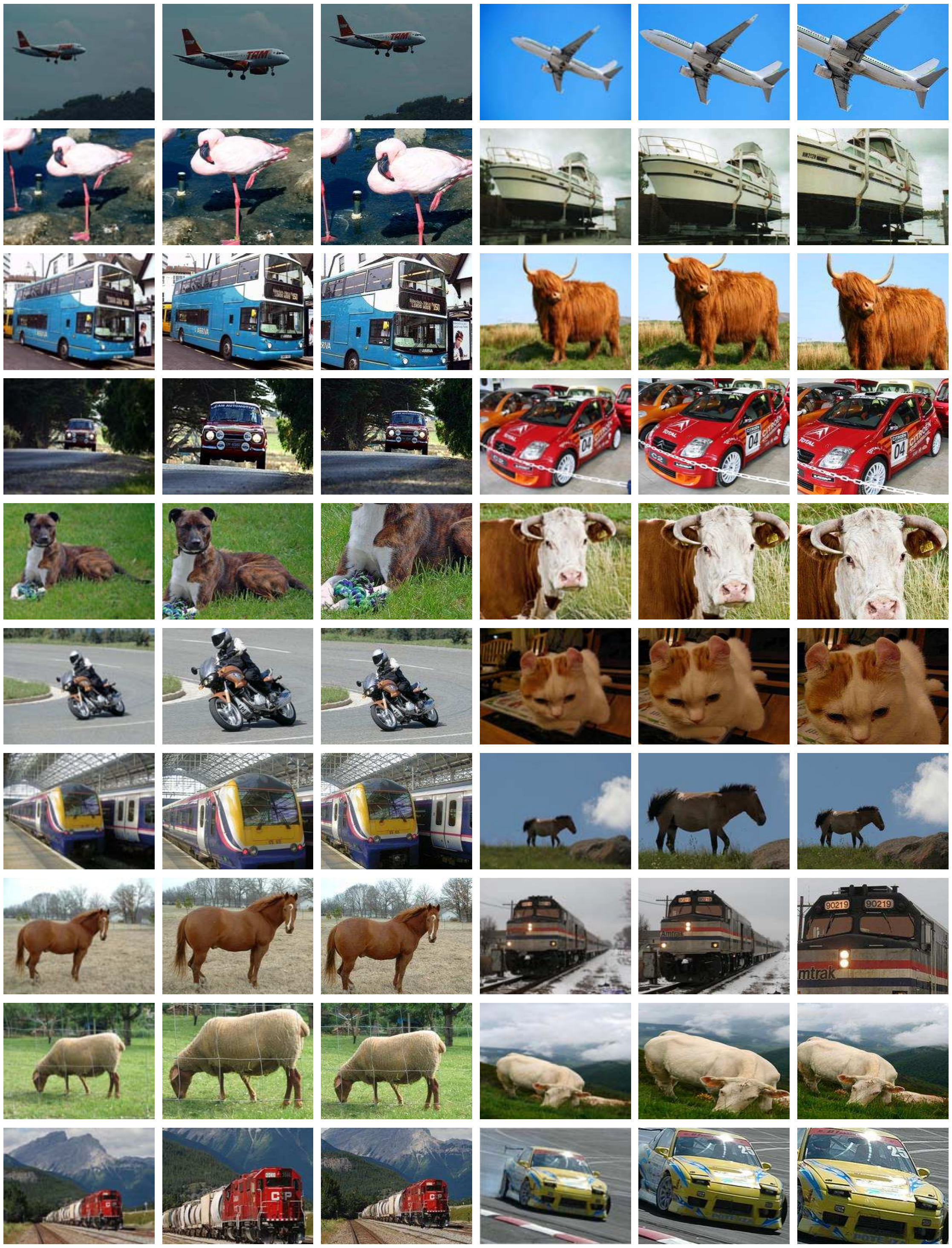


Figure 3.14: Qualitative image retargeting performance comparison between [1] and ours. From left to right: images, our results, results of [1]. Clearly, the performance of our approach is better than that of [1].

Chapter 4

Characterness: an indicator of text in the wild

4.1 Characterness model

As the core component of the proposed scene text detection approach (the pipeline is shown in Fig. 4.1), the characterness model comprises two phases, *i.e.*, candidate region extraction and characterness evaluation. In the first phase, an affine-invariant region detector (modified MSER in our case) is applied to locate potential characters. In the second phase, three novel text-specific features are proposed to model the probability the extracted regions belonging to true characters jointly. Similar to *objectness* in [15], we name the evaluation process as *characterness* evaluation. The details of each phase are as follows.

4.1.1 Candidate region extraction

MSER [52] is an effective region detector which has been applied in various vision tasks, such as tracking [67], image matching [68], and scene text detection [69, 70, 47, 54, 51] amongst others. Roughly speaking, for a gray-scale image, MSERs are those which have a boundary which remains relatively unchanged over a set of different intensity thresholds. The MSER detector is thus particularly well suited to identifying regions with almost uniform intensity surrounded by contrasting background.

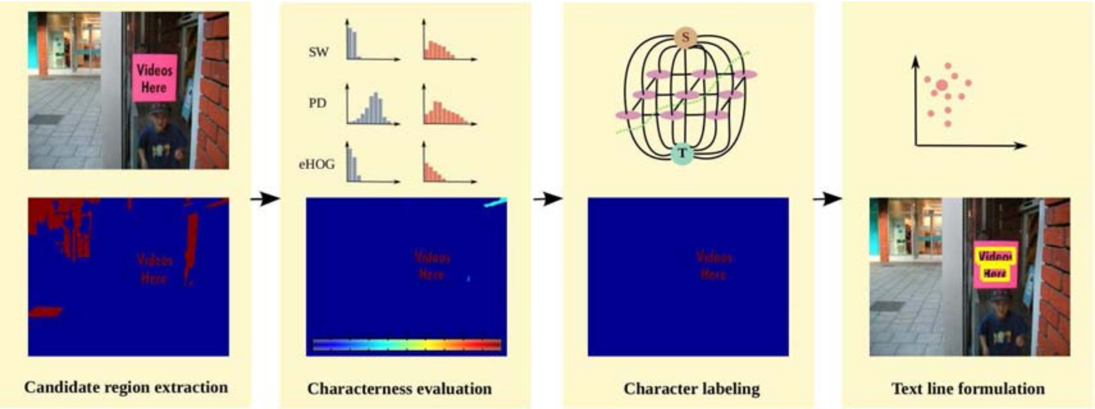
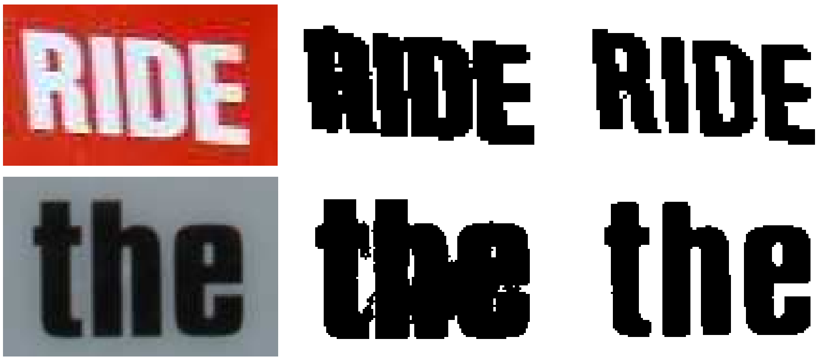


Figure 4.1: Overview of our scene text detection approach. The characterness model consists of the first two phases.

For the task of scene text detection, although the original MSER algorithm is able to detect characters in most cases, there are some characters that are either missed or incorrectly connected (Fig. 4.2 (b)). This tends to degrade the performance of the following steps in the scene text detection algorithms. To address this problem, Chen *et al.* [47] proposed to prune out MSER pixels which were located outside the boundary detected by Canny edge detector. Tsai *et al.* [54] performed judicious parameter selection and multi-scale analysis of MSERs. Neumann and Matas extended MSER to MSER++ [70] and later Extremal Region (ER) [71]. In this work, we use the edge-preserving MSER algorithm from our earlier work [48] (*c.f.* Algorithm 1).

Motivation. As illustrated in some previous work [72, 68], the MSER detector is sensitive to blur. We have observed through empirical testing that this may be attributed to the large quantities of *mixed pixels* (pixels lie between dark background and bright regions, and *vice versa*) present along character boundaries. We notice that these mixed pixels usually have larger gradient amplitude than other pixels. We thus propose to incorporate the gradient amplitude so as to produce edge-preserving MSERs (see Fig. 4.2(c)).



(a) Original text (b) Original MSER (c) Our results

Figure 4.2: Cases that the original MSER fails to extract the characters while the modified eMSER succeeds.

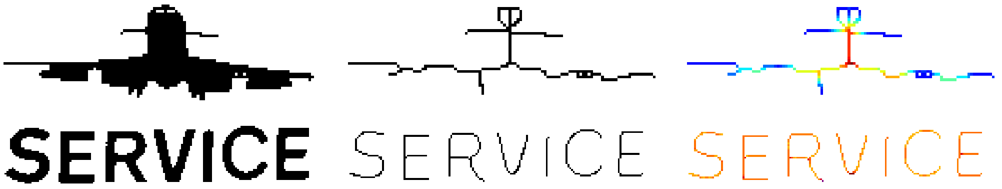
```
Algorithm 1: Edge-preserving MSER (eMSER)
Input: A color image, and required parameters
Output: Regions contain characters and non-characters
Convert the color image to an intensity image $I$.
Smooth $I$ using a guided filter [73].
Compute the gradient amplitude map $\nabla I$, then normalize it to $[0, 255]$.
Get a new image $I^* = I + \gamma \nabla I$ (resp. $I^* = I - \gamma \nabla I$).
Perform MSER algorithm on $I^*$ to extract dark regions on the bright
background (resp. bright regions on the dark background).
```

4.1.2 Characterness evaluation

4.1.2.1 Characterness cues

Characters attract human attention because their appearance differs from that of their surroundings. Here, we propose three novel cues to measure the unique properties of characters.

Stroke Width (SW). Stroke width has been a widely exploited feature for text detection [43, 45, 49, 50]. In particular, SWT [43] computes the length of a straight line between two edge pixels in the perpendicular direction, which is used as a preprocessing step for later algorithms [49, 74, 53]. In [50], a stroke is defined as a connected image region with uniform color and half-closed boundary. Although this assumption is not supported by many uncommon typefaces, stroke



(a) Detected regions (b) Skeleton (c) Distance transform

Figure 4.3: Efficient stroke width computation [2] (best viewed in color). Note the color variation of non-characters and characters on (c). Larger color variation indicates larger stroke width variance.

width remains a valuable cue.

Based on the efficient stroke width computation method we have developed earlier [2] (*c.f.* Algorithm 2), the stroke width cue of region r is defined as:

$$\mathrm{SW}(r) = \frac{\mathrm{Var}(l)}{\mathrm{E}(l)^2}, \tag{4.1}$$

where is $\mathrm{E}(l)$ and $\mathrm{Var}(l)$ are stroke width mean and variance (*c.f.* Algorithm 2). In Fig. 4.3 (c), we use color to visualize the stroke width of exemplar characters and non-characters, where larger color variation indicates larger stroke width variance and *vice versa*.

Algorithm 2: Efficient computation of stroke width

```
Input: A region $r$
Output: Stroke width mean $\mathrm{E}(l)$ and variance $\mathrm{Var}(l)$
Extract the skeleton $S$ of the region.
For each pixel $p \in S$, find its shortest path to the region boundary via
  distance transform. The corresponding length $l$ of the path is defined as
  the stroke width.
Compute the mean $\mathrm{E}(l)$ and variance $\mathrm{Var}(l)$.
```

Perceptual Divergence (PD). As stated in Sec. 2.1, color contrast is a widely adopted measurement of saliency. For the task of scene text detection, we observed that, in order to ensure reasonable readability of text to a human, the color of text in natural scenes is typically distinct from that of the surrounding area. Thus, we propose the PD cue to measure the perceptual divergence of a region r against its surroundings, which is defined as:

$$\mathrm{PD}(r) = \sum_{R,G,B} \sum_{j=1}^{b} h_j(r) \log \frac{h_j(r)}{h_j(r^*)}, \tag{4.2}$$

where the term $\int_x p(x) \log \frac{p(x)}{q(x)}$ is the Kullback-Leibler divergence (KLD) measuring the dissimilarity of two probability distributions in the information theory. Here we take advantage of its discrete form [14], and replace the probability distributions $p(x)$, $q(x)$ by the color histograms of two regions $h(r)$ and $h(r^*)$ (r^* denotes the region outside r but within its bounding box) in a sub-channel respectively. $\{j\}_1^b$ is the index of histogram bins. Note that the more different the two histograms are, the higher the PD is.

In [55], the authors quantified the perceptual divergence as the overlapping areas between the normalized intensity histograms. However, using the intensity channel only ignores valuable color information, which will lead to a reduction in the measured perceptual divergence between distinct colors with the same intensity. In contrast, all three sub-channels (*i.e.*, R, G, B) are utilized in the computation of perceptual divergence in our approach.

Histogram of Gradients at Edges (eHOG). The Histogram of Gradients (HOGs) [75] is an effective feature descriptor which captures the distribution of gradient magnitude and orientation. Inspired by [44], we propose a characterness cue based on the gradient orientation at edges of a region, denoted by eHOG. This cue aims to exploit the fact that the edge pixels of characters typically appear in pairs with opposing gradient directions [44][1].

Firstly, edge pixels of a region r are extracted by the Canny edge detector. Then, gradient orientations θ of those pixels are quantized into four types, *i.e.*, Type 1: $0 < \theta \leq \pi/4$ or $7\pi/4 < \theta \leq 2\pi$, Type 2: $\pi/4 < \theta \leq 3\pi/4$, Type 3: $3\pi/4 < \theta \leq 5\pi/4$, and Type 4: $5\pi/4 < \theta \leq 7\pi/4$. An example demonstrating the four types of edge pixels for text is shown in Fig. 4.4 (right), where four different

[1] Let us assume the gradient orientation of an edge pixel p is θ_p. If we follow the ray along this direction or its inverse direction, we would possibly find another edge pixel q, whose gradient orientation, denoted by θ_q, is approximately opposite to p, *i.e.*, $|\theta_p - \theta_q| \approx \pi$, as edges of a character are typically closed.

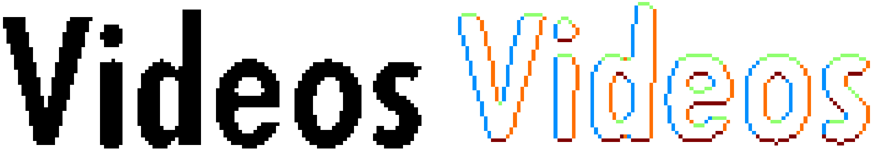


Figure 4.4: Sample text (left) and four types of edge points represented in four different colors (right). Note that the number of edge points in blue is roughly equal to that in orange, and so for green and crimson.

colors are used to depict the four types of edge pixels. As it shows, we can expect that the number of edge pixels in Type 1 should be close to that in Type 3, and so for Type 2 and Type 4.

Based on this observation, we define the eHOG cue as:

$$\mathrm{eHOG(r)} = \frac{\sqrt{(w_1(r) - w_3(r))^2 + (w_2(r) - w_4(r))^2}}{\sum_{i=1}^{4} w_i(r)}, \tag{4.3}$$

where $w_i(r)$ denotes the number of edge pixels in Type i within region r, and the denominator $\sum_{i=1}^{4} w_i(r)$ is for the sake of scale invariance.

4.1.2.2 Bayesian multi-cue integration

The aforementioned cues measure the characterness of a region r from different perspectives. SW and eHOG distinguish characters from non-characters on the basis of their differing intrinsic structures. PD exploits surrounding color information. Since they are complementary and obtained independently of each other, we argue that combining them in the same framework outperforms any of the cues individually.

Following the Naive Bayes model, we assume that each cue is conditionally independent. Therefore, according to Bayes' theorem, the posterior probability that a region is a character (its characterness score) can be computed as:

$$\begin{aligned} p(c|\Omega) &= \frac{p(\Omega|c)p(c)}{p(\Omega)} \\ &= \frac{p(c)\prod_{cue\in\Omega} p(cue|c)}{\sum_{k\in\{c,b\}} p(k)\prod_{cue\in\Omega} p(cue|k)}, \end{aligned}$$

where $\Omega = \{\mathrm{SW}, \mathrm{PD}, \mathrm{eHOG}\}$, and $p(c)$ and $p(b)$ denote the prior probability of characters and background respectively, which we determine on the basis of relative frequency. We model the observation likelihood $p(cue|c)$ and $p(cue|b)$ via distribution of cues on positive and negative samples respectively, with details provided as follows.

Learning the Distribution of Cues. In order to learn the distribution of the proposed cues, we use the training set of text segmentation task in the ICDAR 2013 robust reading competition (challenge 2). To our knowledge, this is is the only benchmark dataset with pixel-level ground truth so far. This dataset contains 229 images harvested from natural scenes. We randomly selected 119 images as the training set, while the rest 100 images were treated as the test set for characterness evaluation in our experiment (Sec. 4.1.2).

- To learn the distribution of cues on positive samples, we directly compute the three cues on characters, as pixel-level ground truth is provided.
- To learn the distribution of cues on negative samples, eMSER algorithm is applied twice to each training image. After erasing ground truth characters, the rest of the extracted regions are considered as negative samples on which we compute the three cues.

Fig. 4.5 shows the distribution of the three cues via normalized histograms. As it shows, for both SW and eHOG, compared with non-characters, characters are more likely to have relatively smaller values (almost within the first 5 bins). For the distribution of PD, it is clear that characters tend to have higher contrast than that of non-characters.

4.2 Labeling and grouping

4.2.1 Character labeling

4.2.1.1 Labeling model overview

We cast the task of separating characters from non-characters as a binary labeling problem. To be precise, we construct a standard graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i\}$

is the vertex set corresponding to the candidate characters, and $\mathcal{E} = \{e_j\}$ is the edge set corresponding to the interaction between vertexes.[1] Each $v_i \in \mathcal{V}$ should be labeled as either character, *i.e.*, $l_i = 1$, or non-character, *i.e.*, $l_i = 0$. Therefore, a labeling set $\mathcal{L} = \{l_i\}$ represents the separation of characters from non-characters. The optimal labeling $\mathcal{L}^*$ can be found by minimizing an energy function:

$$\mathcal{L}^* = \arg\min_{\mathcal{L}} E(\mathcal{L}), \tag{4.4}$$

where $E(\mathcal{L})$ consists of the sum of two potentials:

$$E(\mathcal{L}) = U(\mathcal{L}) + V(\mathcal{L}) \tag{4.5}$$

$$U(\mathcal{L}) = \sum_i u_i(l_i) \tag{4.6}$$

$$V(\mathcal{L}) = \sum_{ij\in\mathcal{E}} v_{ij}(l_i, l_j), \tag{4.7}$$

where $u_i(l_i)$ is the unary potential which determines the cost of assigning the label l_i to v_i. $v_{ij}(l_i, l_j)$ is the pairwise potential which reflects the cost of assigning different labels to v_i and v_j. This model is widely adopted in image segmentation algorithms [76, 77]. The optimal $\mathcal{L}^*$ can be found efficiently using graph-cuts [35] if the pairwise potential is submodular.

4.2.1.2 The design of unary potential

characterness score of extracted regions is encoded in the design of unary potential directly:

$$u_i(l_i) = \begin{cases} p(c|\Omega) & l_i = 0 \\ 1 - p(c|\Omega) & l_i = 1. \end{cases} \tag{4.8}$$

4.2.1.3 The design of pairwise potential

As characters typically appear in homogeneous groups, the degree to which properties of a putative character (stroke width and color, for example) match those

[1] In our work, we consider the edge between two vertexes (regions) exists only if the Euclidean distance between their centroids is smaller than the minimum of their characteristic scales. Characteristic scale is defined as the sum of the length of major axis and minor axis [49].

of its neighbors is an important indicator. This clue plays an important role for human vision to distinguish characters from cluttered background and can be exploited to design the pairwise potential. In this sense, similarity between extracted regions is measured by the following two cues.

Stroke Width Divergence (SWD). To measure the stroke width divergence between two extracted regions r_1 and r_2, we leverage on stroke width histogram. In contrast with Algorithm 2 where only pixels on the skeleton are taken into account, distance transform is applied to all pixels within the region to find length of shortest path. Therefore, the stroke width histogram is defined as the histogram of shortest length. Then, SWD is measured as the discrete KLD (*c.f.* Equ.4.2) of two stroke width histograms.

Color Divergence (CD). The color divergence of two regions is the distance of their average color (in the LAB space) measured by L2 norm.

The aforementioned two cues measure divergence between two regions from two distinct prospectives. Here, we combine them efficiently to produce the unified divergence (UD):

$$\mathrm{UD}(\mathrm{r}_1, \mathrm{r}_2) = \beta\mathrm{SWD}(\mathrm{r}_1, \mathrm{r}_2) + (1-\beta)\mathrm{CD}(\mathrm{r}_1, \mathrm{r}_2), \tag{4.9}$$

where the coefficient β specifies the relative weighting of the two divergence. Without losing generality, in our experiments we set $\beta = 0.5$ so that the two divergence are equally weighted. We make use of the unified divergence to define the pairwise potential as:

$$v_{ij}(l_i, l_j) = [l_i \neq l_j](1 - \tanh(\mathrm{UD}(\mathrm{r_i}, \mathrm{r_j}))), \tag{4.10}$$

where $[\cdot]$ is the Iverson bracket. In other words, the more similar the color and stroke width of the two vertexes are, the less likely they are assigned with different labels.

4.2.2 Text line formulation

The goal of this step, given a set of characters identified in the previous step, is to group them into readable *lines* of text. A comparable step is carried out in some existing text detection approaches [43, 49], but the fact that these methods have many parameters which must be tuned to adapt to individual data means that the adaptability of these methods to various data sets remains unclear. We thus introduce a mean shift based clustering scheme for text line extraction.

Two features exploited in mean shift clustering are characteristic scale and major orientation [49]. Note that both features are normalized. Clusters with at least two elements are retained for further processing.

Within each cluster a bottom-up grouping method is performed, with the goal that only characters within the same line of text will be assigned the same label. In order to achieve this goal all regions are set as unlabeled initially. For an unlabeled region, if another unlabeled region is nearby (less than the average of their skeleton length), both are given the same label and the angle of the line connecting their centroids is taken as the text line angle. On the other hand, for a labeled region, if another unlabeled region is nearby and the angle between them is similar to that of the text line (less than 30 degrees), the latter is assigned the label of the former, and the angle of the text line is updated.

4.3 Evaluation of the characterness model

To demonstrate the effectiveness of the proposed characterness model, we follow the evaluation of salient object detection algorithm in the last chapter. Our characterness map is normalized to [0,1], thus treated as saliency map. Pixels with high saliency value (*i.e.,* intensity) are likely to belong to salient objects (characters in our scenario) which catch human attention.

We qualitatively and quantitatively compare the proposed 'characterness' approach with ten existing saliency detection models: the classical Itti's model (IT) [3], the spectral residual approach (SR) [6], the frequency-tuned approach (FT) [9], context-aware saliency (CA) [19], Zhai's method (LC) [8], histogram-based saliency (HC) [12], region-based saliency (RC) [12], Jiang's method (CB) [18], Rahtu's method (RA) [21] and more recently low-rank matrix decomposition (LR) [17]. Note that CB, RC and CA are considered as the best salient object detection models in the benchmark work [63]. For SR and LC, we use the implementation from [12]. For the rest approaches, we use the publicly available implementations from the original authors. To the best of our knowledge, we are the first to evaluate the state-of-the-art saliency detection models for reflecting characterness in this large quantity.

Unless otherwise specified, three parameters in Algorithm 1 were set as follows: the delta value (Δ) in the MSER was set to 10, and the local window radius in the guided filter was set to 1, $\gamma = 0.5$. We empirically found that these parameters work well for different datasets.

4.3.1 Experimental setup

Datasets For the sake of more precise evaluation of 'characterness', we need pixel-level ground truth of characters.[1] As mentioned in Sec. 4.1.2, to date, the only benchmark dataset with pixel-level ground truth is the training set of text segmentation task in the ICDAR 2013 robust reading competition (challenge 2) which consists of 229 images. Therefore, we randomly selected 100 images of this dataset here for evaluation (the other 119 images have been used for learning the distribution of cues in the Bayesian framework).

Evaluation criteria For a given saliency map, three criteria are adopted to evaluate the quantitative performance of different approaches: precision-recall (PR) curve, F-measure and VOC overlap score. In all three cases we generate a binary segmentation mask of the saliency map at a threshold T.

To obtain the PR curve, we first get 256 binary segmentation masks from the saliency map using threshold T ranging from 0 to 255, as in [9, 12, 17, 16]. For each segmentation mask, precision and recall rate are obtained by comparing it with the ground truth mask. Therefore, in total 256 pairs of precision and recall rates are utilized to plot the PR curve.

In contrast with the computation of the PR curve, to get F-measure [9], T is fixed as the twice of the mean saliency value of the image to get precision rate P and recall rate R. F-measure is computed as $((\beta^2 + 1)P \cdot R)/(\beta^2 P + R)$. We set $\beta^2 = 0.3$ as that in [9].

[1] Dataset of pixel-level ground truth is also adopted in [29]. However, it is not publicly available.

The VOC overlap score [65] is defined as $\frac{|S \cap S'|}{|S \cup S'|}$. Here, S is the ground truth mask, and S' is the our segmentation mask obtained by binarizing the saliency map using the same adaptive threshold T during the calculation of F-measure.

The resultant PR curve (resp. F-measure, VOC overlap score) of a dataset is generated by averaging PR curves (resp. F-measure, VOC overlap score) of all images in the dataset.

4.3.2 Comparison with other approaches

Table 4.1: Performance of all the eleven approaches in VOC overlap scores.

Ours	LR	CB	RC	HC	RA	CA	LC	SR	FT	IT
0.5143	0.2766	0.1667	0.2717	0.2032	0.1854	0.2179	0.2112	0.2242	0.1739	0.1556

As it shows in Fig. 4.6, in terms of the PR curve, all existing saliency detection models, including three best saliency detection models in [63] achieve low precision rate (below 0.5) in most cases when the recall rate is fixed. However, the proposed characterness model produces significantly better results, indicating that our model is more suitable for the measurement of characterness. The straight line segment of our PR curve (when recall rates ranging from 0.67 to 1) is attributed to the fact only foreground regions extracted by eMSER are considered as character candidates, thus background regions always have a zero saliency value. It can also be observed from the PR curve that in our scenario, two best existing saliency detection models are RC and LR.

Precision, recall and F-measure computed via adaptive threshold are illustrated in the Fig. 4.7. Our result significantly outperforms other saliency detection models in all three criteria, which indicates that our approach consistently produces results closer to ground truth.

Table 4.1 illustrates the performance of all approaches measured by VOC overlap score. As it shows, our result is almost twice that of the best saliency detection model LR on this task.

Fig. 4.8 shows some saliency maps of all approaches. It is observed that our approach has obtained visually more feasible results than other approaches have: it usually gives high saliency values to characters while suppressing non-characters, whereas the state-of-the-art saliency detection models may be attracted by other objects in the natural scene (*e.g.*, sign boards are also considered as salient objects in CB).

In summary, both quantitative and qualitative evaluation demonstrate that the proposed characterness model significantly outperforms saliency detection approaches on this task.

4.4 Evaluation of the proposed scene text detection approach

In this section, we evaluate our scene text detection approach as a whole. Same as previous work on scene text detection, we use the detected bounding boxes to evaluate performance and compare with state-of-the-art approaches. Compared with Sec. 4.3 in which only 119 images are utilized to learn the distribution of cues, all images with pixel-level ground truth (229 images) are adopted here, thus the distribution is closer to the real scene.

From the large body of work on scene text detection, we compare our result with some state-of-the-art approaches, including TD method [49], Epshtein's method [43], Li's method [2, 48], Yi's method [46], Meng's method [30], Neumann's method [69, 71], Zhang's method [44] and some approaches presented in the ICDAR competitions. Note that the bandwidth of mean shift clustering in the text line formulation step was set to 2.2 in all experiments.

4.4.1 Experimental setup

Datasets We have conducted comprehensive experimental evaluation on three publicly available datasets. Two of them are from the benchmark ICDAR robust reading competition held in different years, namely ICDAR 2003 [78] and ICDAR 2011 [79]. ICDAR 2003 dataset contains 258 images for training and 251 images for testing. This dataset was also adopted in the ICDAR 2005 [80] competition. ICDAR 2011 dataset contains two folds of data, one for training with 229 images, and the other one for testing with 255 images. To evaluate the effectiveness of the proposed algorithm on text in arbitrary orientations, we also adopted the

Oriented Scene Text Database (OSTD) [46] in our experiments. The dataset set contains 89 images with text lies in in arbitrary orientations.

Evaluation criteria According to literature review, precision, recall and f-measure are the most popularly adopted criteria used to evaluate scene text detection approaches. In general, given the set of correct detection C, total detection D and ground truth G, precision p is defined as the ratio between the number of correct detection $|C|$ and ground truth $|D|$, *i.e.*, $p = |C|/|D|$. Meanwhile, recall r is computed as the number of correct detection $|C|$ divided by the number of ground truth $|G|$, *i.e.*, $r = |C|/|G|$. Obviously, algorithms that overestimate the number of ground truth are punished with a low precision score, whereas those underestimate the number of ground truth are punished with a low recall score. The aim of f-measure f is adopted to combine precision and recall in an uniform measurement, which is defined as: $f = 1/(\alpha/p + (1-\alpha)/r)$. α is a weight controls the relative importance of p and r. $\alpha = 0.5$ gives equal weight to precision and recall. Though the definition of f-measure is always the same, the computation of the precision and recall is slightly different across datasets.

In the ICDAR 2003 and 2005 competition, precision and recall are computed based on a match function. Similar to overlap ratio, the match function between two rectangle bounding boxes r and $r`$ is defined as $m(r, r`) = \frac{|r \cap r`|}{|r \cup r`|}$. Hence, the best match $m(r, R)$ for a bounding box r in a set of bounding boxes R is defined as: $m(r, R) = \max m(r, r`), \forall r` \in R$. Therefore, precision and recall are computed by finding the best match between detected bounding boxes and ground truth bounding boxes:

$$r = \frac{\sum_{r \in G} m(r, D)}{|G|} \tag{4.11}$$

$$p = \frac{\sum_{r \in D} m(r, G)}{|D|}. \tag{4.12}$$

Clearly, in order to achieve high precision and recall rates in this definition, a text detection system should generate accurate bounding boxes of each word, *i.e.*, one-to-one match. In this sense, algorithms which outputs detected text lines composed of several words (*i.e.* many-to-one match) will be penalized sig-

Table 4.2: Evaluation of Bayesian multi-cue integration on the ICDAR 2011 dataset.

Cues	precision	recall	f-measure
Native Bayes model	**0.80**	**0.62**	**0.70**
SVM	0.71	0.42	0.53
SVM-MRF	0.54	0.62	0.58

nificantly. To overcome this drawback, ICDAR 2011 competition adopts the DetEval software [81] which supports one-to-one matches, one-to-many matches and many-to-one matches. For the OSTD dataset, we use the original definition of precision and recall from the authors [46], which are based on computing the size of overlapping areas between $|D|$ and $|G|$. In all three datasets, f-measure is always defined as the harmonic mean of precision and recall.

4.4.2 eMSER versus MSER

Since the proposed characterness cues are computed on regions, the extraction of informative regions is a prerequisite for the robustness of our approach. To demonstrate that the modified eMSER algorithm improves the performance, we compare it with the original MSER algorithm on the ICDAR 2011 dataset. For fair comparison, when learning the distribution of cues on negative samples, we use MSER rather than eMSER to harvest negative samples and then compute the three cues. Other parts of our approach remain fixed.

Using the MSER algorithm achieves a recall of 66%, a precision of 67% and an f-measure of 66%. In comparison, when the eMSER is adopted, the precision rate is boosted significantly (80%), leading to an improved f-measure (70%). This is owing to that eMSER is capable of preserving shape of regions, whereas regions extracted by MSER are more likely to be blurred which makes cues less effective.

4.4.3 Evaluation of Bayesian multi-cue integration

In Eq. 4.4, based on the assumption that the three proposed characterness cues are conditional independent, we fuse them using the Native Bayes model. Here, to show that the simple Native Bayes model is effective in our scenario, we compare

Table 4.3: Evaluation of characterness cues on the ICDAR 2011 dataset.

Cues	precision	recall	f-measure
SW	0.71	0.63	0.67
PD	0.64	0.63	0.63
eHOG	0.58	0.65	0.61
SW+PD	0.78	0.63	0.68
SW+eHOG	0.74	0.63	0.68
PD+eHOG	0.73	0.63	0.67
SW+PD+eHOG	**0.80**	**0.62**	**0.70**
Baseline	0.53	0.67	0.59

it with two other cue integration configurations:

- SVM. In this configuration, we simply concatenate the three cues to produce a final three-dimensional feature vector for each potential text region. Then, we train a linear SVM classifier after feature normalization. Text regions which are classified as negative are directly removed which means we do not use MRF for character labeling in this case.

- SVM-MRF. In this configuration, as the former one, we still train a linear SVM classifier using concatenated cues. However, instead of using SVM for classification directly, we use the decision value of the SVM output to replace characterness score in the MRF model while the pairwise potential is fixed.

We report the experimental results on the ICDAR 2011 dataset in Table. 4.2. As it shows, whereas both the SVM and SVM-MRF configurations suffer low recall and precision rate respectively, the simple Native Bayes model achieves significantly superior performance than both.

4.4.4 Evaluation of characterness cues

The proposed characterness cues (*i.e.* SW, PD and eHOG) are critical to the characternss model and the final text detection result. In order to show that they are effective in distinguishing characters and non-characters, we evaluate different combinations of the cues on the ICDAR 2011 dataset. Table 4.3 shows the evaluation via precision, recall and f-measure. Clearly, the table shows an upward trend in performance with increasing number of cues. Note that the baseline method in Table 4.3 corresponds to the result obtained by directly preforming text line formulation after candidate region extraction.

As it shows in Table 4.3, the performance of the proposed approach is generally poorer when only one cue is adopted. However, the f-measures are still much higher than the baseline method, which indicates that individual cues are effective. We also notice that the SW cue shows the best f-measure when individual cue is considered. This indicates that characters and non-characters are much easier to be separated by using the SW cue. From Table 4.3, we can easily conclude that the order of discriminability of individual cues (from high to low) is: SW, PD and eHOG.

The performance of the proposed approach is boosted by a large extent (about 5% in f-measure) when two cues are combined, which attributes to the significant increase in precision.

As expected, the pest performance is achieved when all cues are combined. Although there is a slightly drop in recall rate, precision rate (80%) is significantly higher than all other combinations, thus the f-measure is the best.

4.4.5 Comparison with other approaches

Table 4.4 and Table 4.5 show the performance of our approach on two benchmark datasets (*i.e.* ICDAR 2003 and 2011), along with the performance of other state-of-the-art scene text detection algorithms. Note that methods without reference correspond to those presented in each competition.

On the ICDAR 2003 dataset, our method achieves significantly better precision (79%) than all other approaches. Besides, our recall rate (64%) is above the average, thus our f-measure (71%) is superior than others. Although supervised learning (random forest) is adopted in TD-Mixture [49], its precision (69%) is much lower than ours (79%), which indicates the strong discriminability of the Bayesian classifier which is based on fusion of characterness cues.

On the ICDAR 2011 dataset, our method achieves a precision of 80%, a recall of 62%, and an f-measure of 70%. In terms of precision, our rate (80%) is only 1% lower than that of Kim's method [51] (81%) which is based on two times of

Table 4.4: Results on ICDAR 2003 dataset.

method	precision	recall	f-measure
Ours	**0.79**	**0.64**	**0.71**
Kim [51]	0.78	0.65	0.71
TD-Mixture [49]	0.69	0.66	0.67
Yi [50]	0.73	0.67	0.66
Epshtein [43]	0.73	0.60	0.66
Li [48]	0.62	0.65	0.63
Yi [46]	0.71	0.62	0.62
Becker [80]	0.62	0.67	0.62
Meng [30]	0.66	0.57	0.61
Li [2]	0.59	0.59	0.59
Chen [80]	0.60	0.60	0.58
Neumann [69]	0.59	0.55	0.57
Zhang [44]	0.67	0.46	0.55
Ashida	0.55	0.46	0.50

Table 4.5: Results on ICDAR 2011 dataset.

method	precision	recall	f-measure
Kim [51]	0.81	0.69	0.75
Ours	**0.80**	**0.62**	**0.70**
Neumann [71]	0.73	0.65	0.69
Li [48]	0.63	0.68	0.65
Yi	0.67	0.58	0.62
TH-TextLoc	0.67	0.58	0.62
Li [2]	0.59	0.62	0.61
Neumann	0.69	0.53	0.60
TDM_IACS	0.64	0.54	0.58
LIP6-Retin	0.63	0.50	0.56
KAIST AIPR	0.60	0.45	0.51
ECNU-CCG	0.35	0.38	0.37
Text Hunter	0.50	0.26	0.34

supervised learning. Besides, we report the best performance amongst all region-based approaches.

Our method achieves a precision of 72%, a recall of 60% and an f-measure of 61% on the OSTD dataset [46] whcih outperforms Yi's method [46], with an improvement of 6% in f-measure.

Fig. 4.9 shows some sample outputs of our method with detected text bounded by yellow rectangles. As it shows, our method can handle several text variations, including color, orientation and size. The proposed method also works well in a wide range of challenging conditions, such as strong light, cluttered scenes, flexible surfaces and so forth.

In terms of failure cases (see Fig. 4.10), there are two culprits of false negatives. Firstly, the candidate region extraction step misses some characters with very low resolution. Furthermore, some characters in uncommon fonts are likely to have low characterness score, thus likely to be labeled as non-characters in the character labeling model. This problem may be solved by enlarging the training sets to get more accurate distribution of characterness cues. On the other hand, most false positives stem from non-characters whose distribution of cues is similar to that of characters.

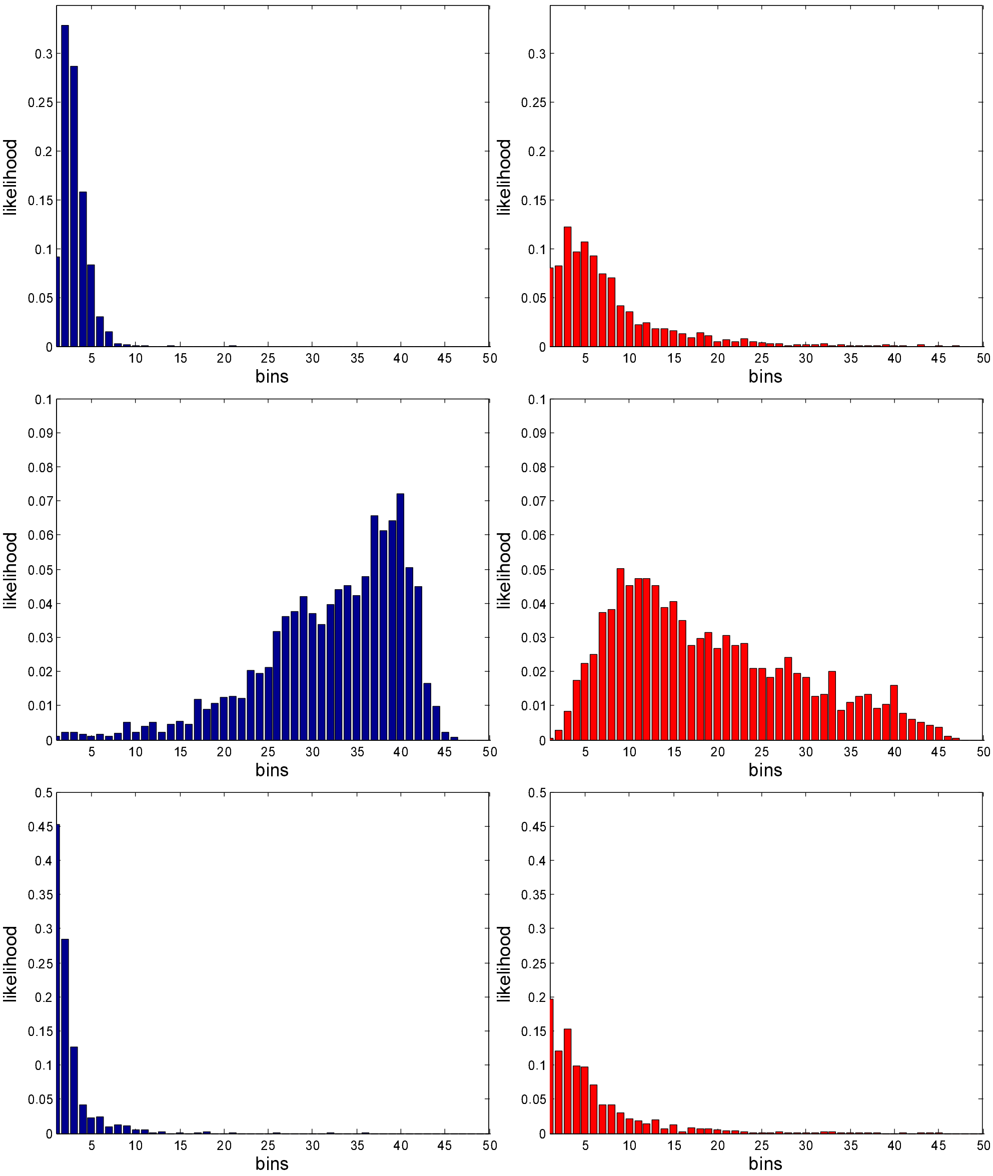


Figure 4.5: Observation likelihood of characters (blue) and non-characters (red) on three characterness cues *i.e.*, SW (top row), PD (middle row), and eHOG (bottom row). Clearly, for all three cues, observation likelihoods of characters are quite different from those of non-characters, indicating that the proposed cues are effective in distinguishing them. Notice that 50 bins are adopted.

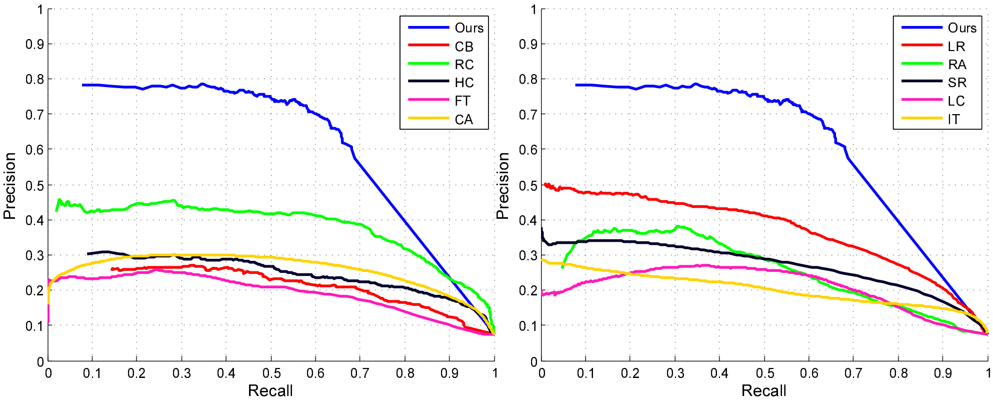


Figure 4.6: Quantitative precision-recall curves performance of all the eleven approaches. Clearly, our approach achieves significant improvement compared with state-of-the-art saliency detection models for the measurement of 'characterness'.

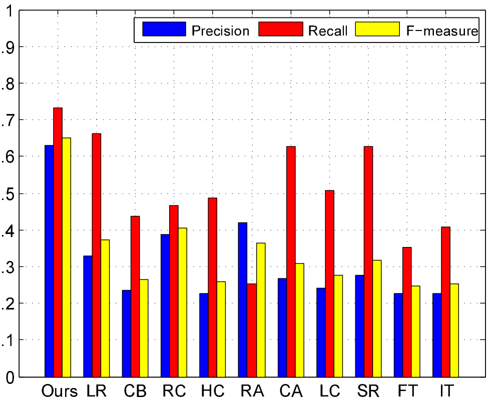


Figure 4.7: Quantitative F-measure performance of all the eleven approaches. Clearly, our approach achieves significant improvement compared with state-of-the-art saliency detection models for the measurement of 'characterness'.

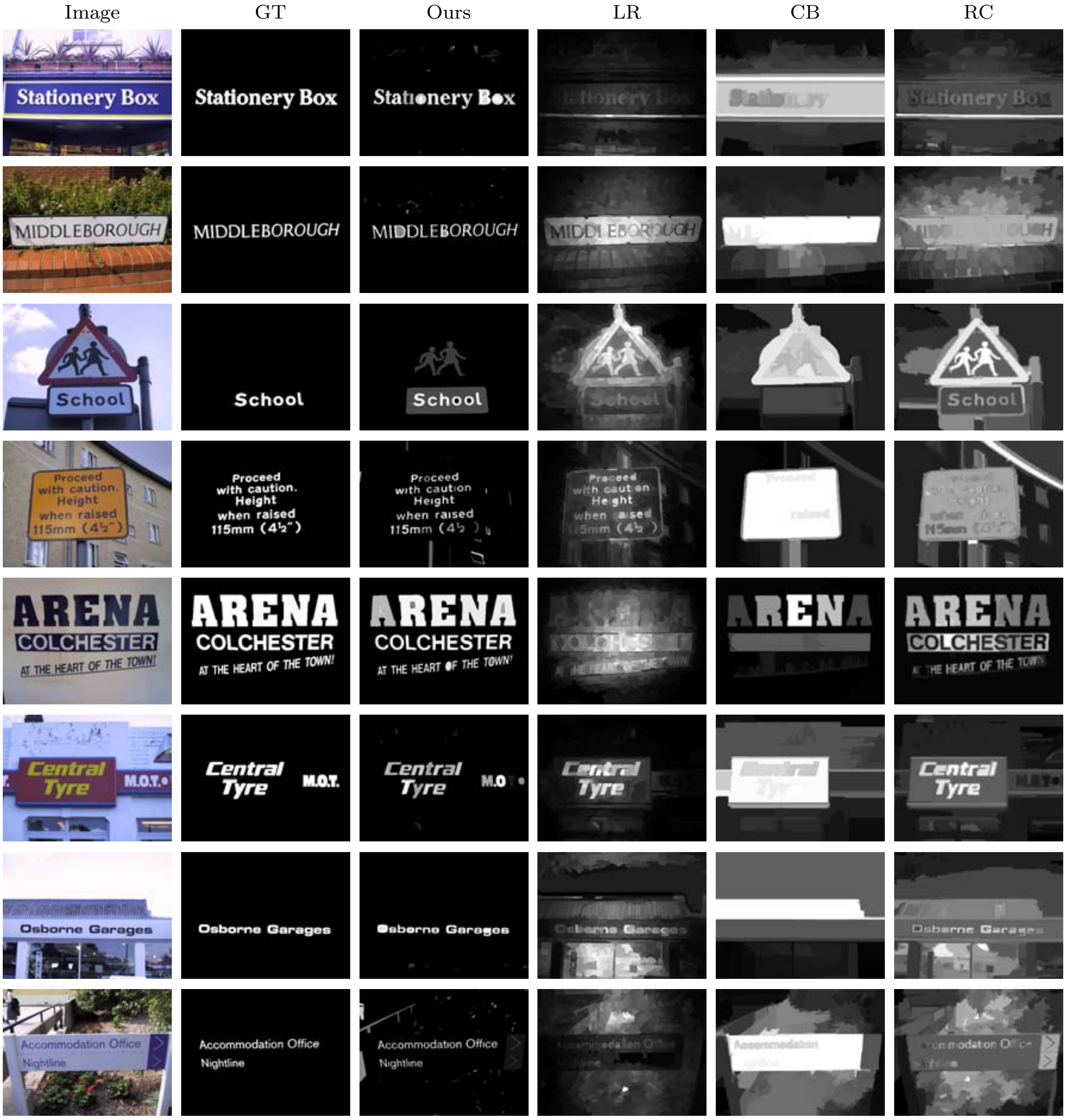


Figure 4.8: Visual comparison of saliency maps. Clearly, the proposed method highlights characters as salient regions whereas state-of-the-art saliency detection algorithms may be attracted by other stuff in the scene.

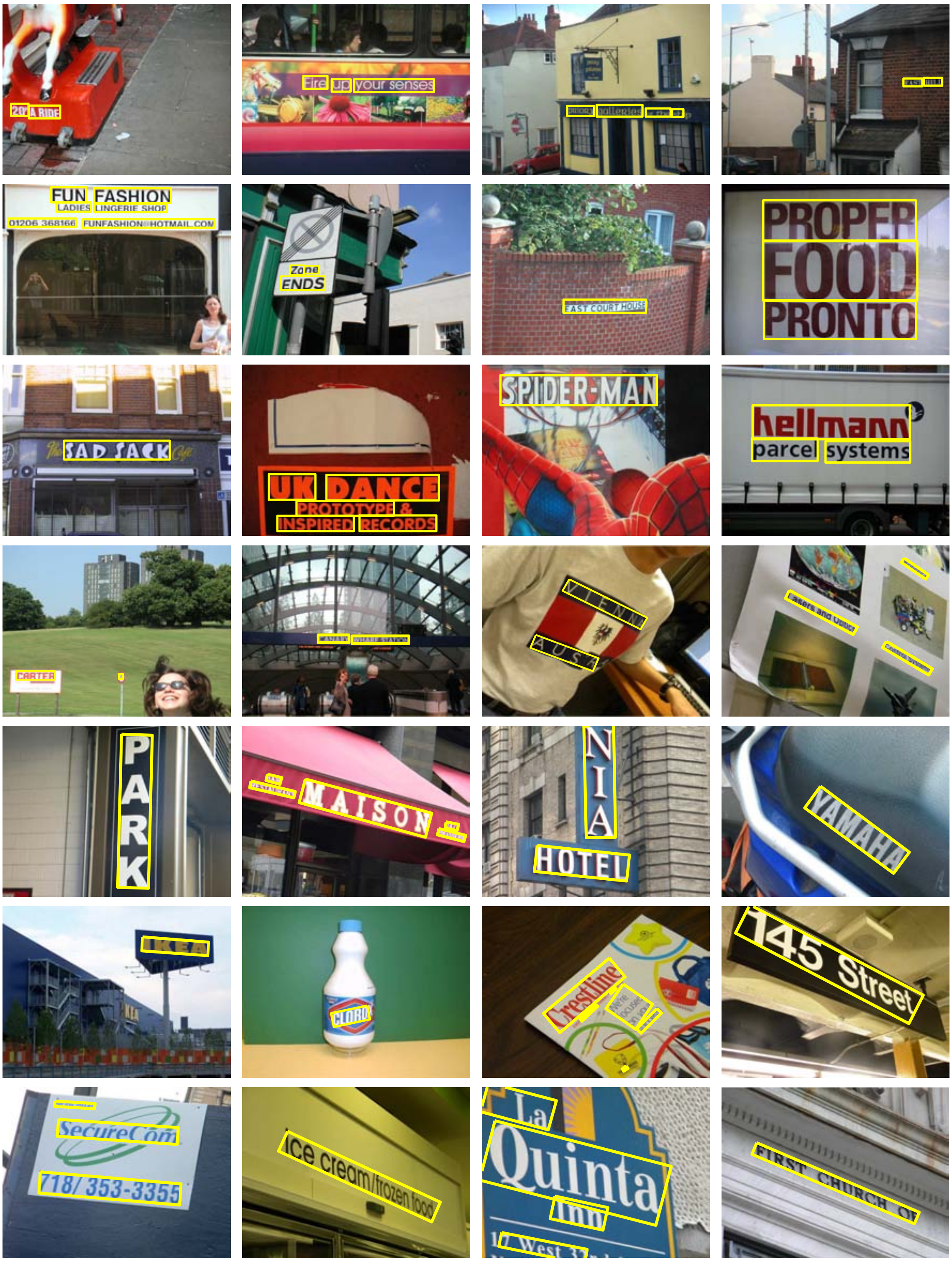


Figure 4.9: Sample outputs of our method on the ICDAR datasets and OSTD dataset. Detected text are in yellow rectangles.

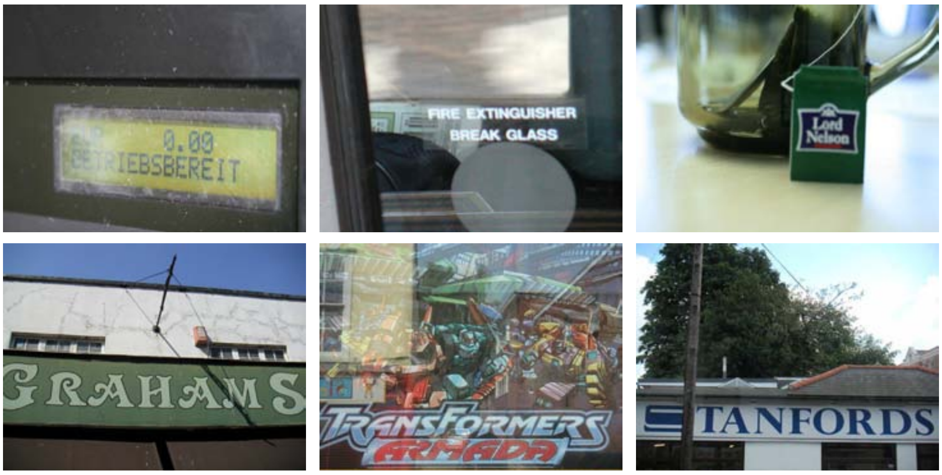


Figure 4.10: False negatives of our approach. Clearly, there are two kinds of characters that our approach cannot handle, (i) characters in extremely blur and low resolution (top row), (ii) characters in uncommon fonts (bottom row).

Chapter 5

Conclusions

This thesis has presented a detailed study of two important topics in the vision community, *saliency detection* and *scene text detection*. For computers, saliency detection aims to locate objects catch human attention, which is a mechanism embedded in the human vision system, while scene text detection aims to locate text in the natural scene. As most previous work study these two subjects separately, this thesis is a pioneering work on studying saliency and scene text detection jointly.

For saliency detection, we have proposed two salient object detection approaches based on hypergraph modeling and center-versus-surround max-margin learning. Specifically, we have designed a hypergraph modeling approach that captures the intrinsic contextual saliency information on image pixels/superpixels by detecting salient vertices and hyperedges in a hypergraph. Furthermore, we have developed a local salient object detection approach based on center-versus-surround max-margin learning, which solves an imbalanced cost-sensitive SVM optimization problem. Compared with the twelve state-of-the-art approaches, we have empirically shown that the fusion of our approaches is able to achieve more accurate and robust results of salient object detection.

From previous observation in a large body of literature which confirms text is an important attribute of human attention, we investigated whether scene text detection could be aided by saliency detection approaches. To achieve this goal, we have proposed a scene text detection approach based on measuring 'characterness'. The proposed characterness model reflects the probability of extracted regions belonging to character, which is constructed via fusion of novel characterness cues in the Bayesian framework. We have demonstrated that this model significantly outperforms the state-of-the-art saliency detection approaches in the task of measuring the ‘characterness’ of text. In the character labeling model, by constructing a standard graph, not only characterness score of individual regions is considered, similarity between regions is also adopted as the pairwise potential. Compared with state-of-the-art scene text detection approaches, we have shown that our method is able to achieve more accurate and robust results of scene text detection.

References

[1] J. Sun and H. Ling, "Scale and object aware image retargeting for thumbnail browsing," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2011, pp. 1511–1518. xi, 1, 29, 30, 36

[2] Y. Li and H. Lu, "Scene text detection via stroke width," in *Proc. IEEE Int. Conf. Patt. Recogn.*, 2012, pp. 681–684. xi, 12, 40, 49, 54

[3] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998. 1, 3, 9, 14, 15, 29, 30, 46

[4] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 155–162. 1

[5] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 545–552. 1

[6] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2007, pp. 1–8. 1, 9, 10, 46

[7] T. Judd, K. A. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2009, pp. 2106–2113. 1, 3

[8] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *ACM Multimedia*, 2006, pp. 815–824. 1, 46

[9] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, “Frequency-tuned salient region detection,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2009, pp. 1597–1604. 1, 2, 9, 10, 23, 29, 30, 46, 47

[10] Y. Wei, F. Wen, W. Zhu, and J. Sun, “Geodesic saliency using background priors,” in *Proc. Eur. Conf. Comp. Vis.*, 2012, pp. 29–42. 1, 2, 8, 9, 10, 21, 29, 30

[11] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, “Salient object detection by composition,” in *Proc. IEEE Int. Conf. Comp. Vis.*, 2011, pp. 1028–1035. 1, 2, 8, 9, 10

[12] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, “Global contrast based salient region detection,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2011, pp. 409–416. 1, 2, 8, 9, 23, 29, 30, 46, 47

[13] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, “Learning to detect a salient object,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, 2011. 1, 2, 3, 8, 14, 23

[14] D. A. Klein and S. Frintrop, “Center-surround divergence of feature statistics for salient object detection,” in *Proc. IEEE Int. Conf. Comp. Vis.*, 2011, pp. 2214–2219. 1, 2, 8, 9, 15, 41

[15] B. Alexe, T. Deselaers, and V. Ferrari, “What is an object?” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2010, pp. 73–80. 1, 2, 3, 4, 8, 37

[16] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, “Saliency filters: Contrast based filtering for salient region detection,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 733–740. 1, 2, 8, 9, 10, 23, 29, 30, 47

[17] X. Shen and Y. Wu, “A unified approach to salient object detection via low rank matrix recovery,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 853–860. 1, 2, 8, 9, 10, 23, 29, 30, 46, 47

[18] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li, “Automatic salient object segmentation based on context and shape prior,” in *Proc. Brit. Mach. Vis. Conf.*, vol. 3, no. 4, 2011, p. 7. 1, 2, 8, 9, 15, 21, 29, 30, 46

[19] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2010, pp. 2376–2383. 1, 2, 8, 9, 29, 30, 46

[20] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2011, pp. 914–921. 1, 2, 8, 29, 30

[21] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting salient objects from images and videos," in *Proc. Eur. Conf. Comp. Vis.*, 2010, pp. 366–379. 1, 8, 9, 15, 29, 30, 46

[22] Y. Ding, J. Xiao, and J. Yu, "Importance filtering for image retargeting," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2011, pp. 89–96. 1

[23] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 3506–3513. 1

[24] L. Wang, J. Xue, N. Zheng, and G. Hua, "Automatic salient object extraction with contextual cue," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2011, pp. 105–112. 1

[25] D. Gao, V. Mahadevan, and N. Vasconcelos, "The discriminant center-surround hypothesis for bottom-up saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 497–504. 1

[26] X. Hou and L. Zhang, "Dynamic visual attention: searching for coding length increments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 681–688. 1, 29, 30

[27] M. Cerf, E. P. Frady, and C. Koch, "Faces and text attract gaze independent of the task: Experimental data and computer model," *J. Vis.*, vol. 9, no. 12, 2009. 3

[28] Q. Sun, Y. Lu, and S. Sun, "A visual attention based approach to text extraction," in *Proc. IEEE Int. Conf. Patt. Recogn.*, 2010, pp. 3991–3995. 3, 5, 14

[29] A. Shahab, F. Shafait, A. Dengel, and S. Uchida, "How salient is scene text?" in *Proc. IEEE Int. Workshop. Doc. Anal. Syst.*, 2012, pp. 317–321. 3, 5, 14, 47

[30] Q. Meng and Y. Song, "Text detection in natural scenes with salient region," in *Proc. IEEE Int. Workshop. Doc. Anal. Syst.*, 2012, pp. 384–388. 3, 5, 14, 49, 54

[31] S. Uchida, Y. Shigeyoshi, Y. Kunishige, and Y. Feng, "A keypoint-based approach toward scenery character detection," in *Proc. IEEE Int. Conf. Doc. Anal. and Recogn.*, 2011, pp. 819–823. 3, 5, 14

[32] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering occlusion boundaries from an image," *Int. J. Comp. Vis.*, vol. 91, no. 3, pp. 328–346, 2011. 3

[33] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. D. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 3378–3385. 3

[34] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 3282–3289. 3

[35] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001. 4, 6, 13, 44

[36] Z. Jiang and L. S.Davis, "Submodular salient region detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2013, pp. 2043–2050. 8

[37] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. L. Yuille, and C. Koch, "Adaboost for text detection in natural scene," in *Proc. IEEE Int. Conf. Doc. Anal. and Recogn.*, 2011, pp. 429–434. 10, 11

[38] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2004, pp. 366–373. 10, 11

[39] M. Zhao, S. Li, and J. T.-Y. Kwok, "Text detection in images using sparse representation with discriminative dictionaries," vol. 28, no. 12, pp. 1590–1599, 2010. 10, 11

[40] W. Pan, T. D. Bui, and C. Y. Suen, "Text detection from scene images using sparse representation," in *Proc. IEEE Int. Conf. Patt. Recogn.*, 2008, pp. 1–5. 10, 11

[41] 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 *Proc. IEEE Int. Conf. Doc. Anal. and Recogn.*, 2011, pp. 440–445. 10, 11

[42] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. IEEE Int. Conf. Patt. Recogn.*, 2012, pp. 3304–3308. 10, 11

[43] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2010, pp. 2963–2970. 12, 39, 46, 49, 54

[44] J. Zhang and R. Kasturi, "Text detection using edge gradient and graph spectrum," in *Proc. IEEE Int. Conf. Patt. Recogn.*, 2010, pp. 3979–3982. 12, 41, 49, 54

[45] ——, "Character energy and link energy-based text extraction in scene images," in *Proc. Asian Conf. Comp. Vis.*, 2010, pp. 308–320. 12, 13, 39

[46] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Trans. Image Proc.*, vol. 20, no. 9, pp. 2594–2605, 2011. 12, 13, 49, 50, 51, 54, 55

[47] 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 *Proc. IEEE Int. Conf. Image Process.*, 2011, pp. 2609–2612. 12, 37, 38

[48] Y. Li, C. Shen, W. Jia, and A. van den Hengel, “Leveraging surrounding context for scene text detection,” in *Proc. IEEE Int. Conf. Image Process.*, 2013, pp. 2264–2268. 12, 13, 38, 49, 54

[49] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, “Detecting texts of arbitrary orientations in natural images,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 1083–1090. 12, 39, 44, 46, 49, 53, 54

[50] C. Yi and Y. Tian, “Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification,” *IEEE Trans. Image Proc.*, vol. 21, no. 9, pp. 4256–4268, 2012. 12, 13, 39, 54

[51] H. Koo and D. Kim, “Scene text detection via connected component clustering and non-text filtering.” *IEEE Trans. Image Proc.*, vol. 22, no. 6, pp. 2296–2305, 2013. 12, 13, 37, 53, 54

[52] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions,” in *Proc. Brit. Mach. Vis. Conf.*, 2002, pp. 384–393. 12, 37

[53] J. Pan, Y. Chen, B. Anderson, P. Berkhin, and T. Kanade, “Effectively leveraging visual context to detect texts in natural scenes,” in *Proc. Asian Conf. Comp. Vis.*, 2012. 12, 39

[54] S. Tsai, V. Parameswaran, J. Berclaz, R. Vedantham, R. Grzeszczuk, and B. Girod, “Design of a text detection system via hypothesis generation and verification,” in *Proc. Asian Conf. Comp. Vis.*, 2012. 12, 13, 37, 38

[55] Z. Liu and S. Sarkar, “Robust outdoor text detection using text intensity and shape features,” in *Proc. IEEE Int. Conf. Patt. Recogn.*, 2008, pp. 1–4. 13, 41

[56] Y.-F. Pan, X. Hou, and C.-L. Liu, “A hybrid approach to detect and localize texts in natural scene images,” *IEEE Trans. Image Proc.*, vol. 20, no. 3, pp. 800–813, 2011. 13

[57] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, no. 1, pp. 85–105, 2002. 16

[58] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2011, pp. 89–96. 17, 24

[59] D. Zhou, J. Huang, and B. Scholköpf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007. 18

[60] X. Yuan, B. Hu, and R. He, "Agglomerative mean-shift clustering," *IEEE Trans. on Knowledge and Data Engineering*, vol. 24, no. 2, pp. 209–219, 2012. 21

[61] V. Movahedi and J. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. Workshops*, 2010, pp. 49–56. 23

[62] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2007, pp. 1–8. 23

[63] A. Borji, D. N. Sihite, and L. Itti, "Salient object detection: A benchmark," in *Proc. Eur. Conf. Comp. Vis.*, 2012, pp. 414–429. 23, 46, 48

[64] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 996–1010, 2013. 23

[65] A. Rosenfeld and D. Weinshall, "Extracting foreground masks towards object recognition," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2011, pp. 1371–1378. 24, 48

[66] X. Ren and L. Bo, "Discriminatively trained sparse code gradients for contour detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 593–601. 24

[67] M. Donoser and H. Bischof, “Efficient maximally stable extremal region (mser) tracking,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2006, pp. 553–560. 37

[68] P.-E. Forssén and D. G. Lowe, “Shape descriptors for maximally stable extremal regions,” in *Proc. IEEE Int. Conf. Comp. Vis.*, 2007, pp. 1–8. 37, 38

[69] L. Neumann and J. Matas, “A method for text localization and recognition in real-world images,” in *Proc. Asian Conf. Comp. Vis.*, 2010, pp. 770–783. 37, 49, 54

[70] ——, “Text localization in real-world images using efficiently pruned exhaustive search,” in *Proc. IEEE Int. Conf. Doc. Anal. and Recogn.*, 2011, pp. 687–691. 37, 38

[71] ——, “Real-time scene text localization and recognition,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 3538–3545. 38, 49, 54

[72] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. J. V. Gool, “A comparison of affine region detectors,” *Int. J. Comp. Vis.*, vol. 65, no. 1-2, pp. 43–72, 2005. 38

[73] K. He, J. Sun, and X. Tang, “Guided image filtering,” in *Proc. Eur. Conf. Comp. Vis.*, 2010, pp. 1–14. 39

[74] N. B. Ali Mosleh and and A. B. Hamza, “Image text detection using a bandlet-based edge detector and stroke width transform,” in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–12. 39

[75] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2005, pp. 886–893. 41

[76] Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images,” in *Proc. IEEE Int. Conf. Comp. Vis.*, 2001, pp. 105–112. 44

[77] D. Küttel and V. Ferrari, “Figure-ground segmentation by transferring window masks,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 558–565. 44

[78] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, “Icdar 2003 robust reading competitions,” in *Proc. IEEE Int. Conf. Doc. Anal. and Recogn.*, 2003, pp. 682–687. 49

[79] A. Shahab, F. Shafait, and A. Dengel, “Icdar 2011 robust reading competition challenge 2: Reading text in scene images,” in *Proc. IEEE Int. Conf. Doc. Anal. and Recogn.*, 2011, pp. 1491–1496. 49

[80] S. M. Lucas, “Text locating competition results,” in *Proc. IEEE Int. Conf. Doc. Anal. and Recogn.*, 2005, pp. 80–85. 49, 54

[81] C. Wolf and J.-M. Jolion, “Object count/area graphs for the evaluation of object detection and segmentation algorithms,” *Int. J. Doc. Anal. Recogn.*, vol. 8, no. 4, pp. 280–296, 2006. 51