EXTRACTING GENERIC TEXT INFORMATION FROM IMAGES

A Thesis Submitted for the Degree of
Doctor of Philosophy

By

Chao Zeng

in

School of Computing and Communications
UNIVERSITY OF TECHNOLOGY, SYDNEY
AUSTRALIA

SEPTEMBER 2013

UNIVERSITY OF TECHNOLOGY, SYDNEY
SCHOOL OF COMPUTING AND COMMUNICATIONS

The undersigned hereby certify that they have read this thesis entitled "**EXTRACTING GENERIC TEXT INFORMATION FROM IMAGES**" by **Chao Zeng** and that in their opinions it is fully adequate, in scope and in quality, as a thesis for the degree of **Doctor of Philosophy**.

Dated: September 2013

Research Supervisors: ____________________
Xiangjian He

____
Wenjing Jia

CERTIFICATE

Date: **September 2013**

Author: **Chao Zeng**

Title: **EXTRACTING GENERIC TEXT INFORMATION FROM IMAGES**

Degree: **Ph.D.**

I certify that this thesis has not already been submitted for any degree and is not being submitted as part of candidature for any other degree.

I also certify that the thesis has been written by me and that any help that I have received in preparing this thesis, and all sources used, have been acknowledged in this thesis.

Signature of Author

Acknowledgements

First and foremost, I sincerely appreciate my principal supervisor Professor Xiangjian He for providing me with this precious opportunity of studying PhD under his supervision at University of Technology, Sydney (UTS). His insightful guidance and his continuous encouragement give me impetus throughout my entire PhD study. I owe my research achievements to his excellent supervision.

I also would like to express my deepest gratitude to my co-supervisor Dr. Wenjing Jia who always offers enlightening suggestions and patiently corrects my paper writing. Her consistent support during my research work and the completion of this thesis will never be forgotten.

I am also much indebted to the staff and my fellow research students in Faculty of Engineering and Information Technology (FEIT), UTS, especially the following people for offering various assistance during the completion of this research work. They are Qiang Wu, Ruo Du, Muhammad Abul Hasan, Sheng Wang, Man To Wong, Massimo Piccardi, Richard Yi Da Xu, Min Xu, Zhiyuan Tan, Aruna Jamdagni, Liangfu Lu, Jie Liang, Ava Bargi and Damith Mudugamuwa.

My special thanks should extend to my father Xiangheng Zeng, mother Likuan Zhang and my wife Yaxin Xu. This thesis could not have been completed successfully without their persistent encouragement and firm support.

Last but not least, I appreciate the financial assistance provided by International Postgraduate Research Scholarship of Australia and UTS President's Scholarship. Furthermore, FEIT, UTS is also acknowledged for offering me a travel fund for attending an international conference.

To My Parents and My Family

Table of Contents

List of Tables

List of Figures

Abstract

As a vast amount of text appears everywhere, including natural scene, web pages and videos, text becomes very important information for different applications. Extracting text information from images and video frames is the first step of applying them to a specific application and this task is completed by a text information extraction (TIE) system. TIE consists of text detection, text binarisation and text recognition. For different applications or projects, one or more of these three TIE components may be embedded. Although many efforts have been made to extract text from images and videos, this problem is far from being solved due to the difficulties existing in different scenarios. This thesis focuses on the research of text detection and text binarisation.

For the work on text detection in born-digital images, a new scheme for coarse text detection and a texture-based feature for fine text detection are proposed. In the coarse detection step, a novel scheme based on Maximum Gradient Difference (MGD) response of text lines is proposed. MGD values are classified into multiple clusters by a clustering algorithm to create multiple layer images. Then, the text line candidates are detected in different layer images. An SVM classifier trained by a novel texture-based feature is utilized to filter out the non-text regions. The superiority of the proposed feature is demonstrated by comparing with other features for text/non-text classification capability.

Another algorithm is designed for detecting texts from natural scene images. Maximally Stable Extremal Regions (MSERs) as character candidates are classified into character MSERs and non-character MSERs based on geometry-based, stroke-based, HOG-based and colour-based features. Two types of misclassified character MSERs are retrieved by two different schemes respectively. A false alarm elimination step is performed for increasing the text detection precision and the bootstrap strategy is used to enhance the power of suppressing false positives. Both promising recall rate and precision rate are achieved.

In the aspect of text binarisation research, the combination of the selected colour channel image and graph-based technique are explored firstly. The colour channel image with the histogram having the biggest distance, estimated by mean-shift procedure, between the two main peaks is selected before the graph model is constructed. Then, Normalised cut is employed on the graph to get the binarisation result. For circumventing the drawbacks of the grayscale-based method, a colour-based text binarisation method is proposed. A modified Connected Component (CC)-based validation measurement and a new objective segmentation evaluation criterion are applied as sequential processing. The experimental results show the effectiveness of our text binarisation algorithms.

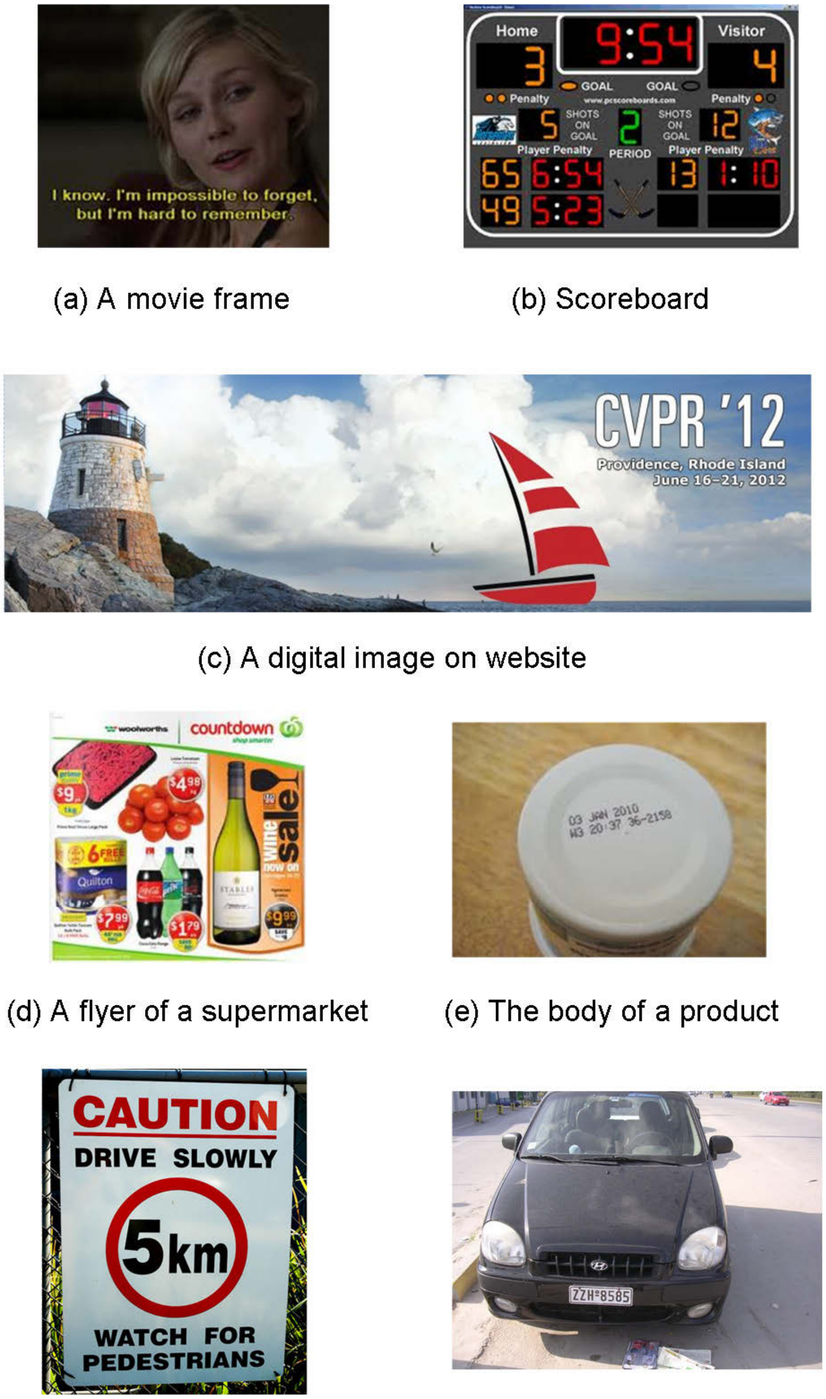
Chapter 1

Introduction

1.1 Applications of Text Information Extraction (TIE) Research

Text, as a carrier of presenting languages, appears at every corner of the world. The subtitles of a movie (Figure 1.1(a)) make audience easier to understand the contents and can translate the original dialogues into understandable text. The scoreboard (Figure 1.1(b)) illustrated in a live soccer match can let the fans know the performance of the two teams. The digital images (Figure 1.1(c)) on the home page of a website always show some textual information to attract the attention of an internet surfer. The prices on a flyer (Figure 1.1(d)) bring an impressive idea of what will be on sale in a supermarket. The text on the body of a product (Figure 1.1(e)) tells the expiry date. Traffic signs (Figure 1.1(f)) provide important directions to pedestrians for the purpose of safety. Letters and Arabic numbers on a vehicle license plate (Figure 1.1(g)) gives the unique identification of the vehicle.

Text information extraction is the terminology representing the process that extracts text contents from still images and image sequences, and then turn them into machine-editable text. A complete text information extraction (TIE) system includes the following three



(a) A movie frame

(b) Scoreboard

(c) A digital image on website

(d) A flyer of a supermarket

(e) The body of a product

(f) A traffic sign

(g) A vehicle license plate on a car

Figure 1.1: Text examples.

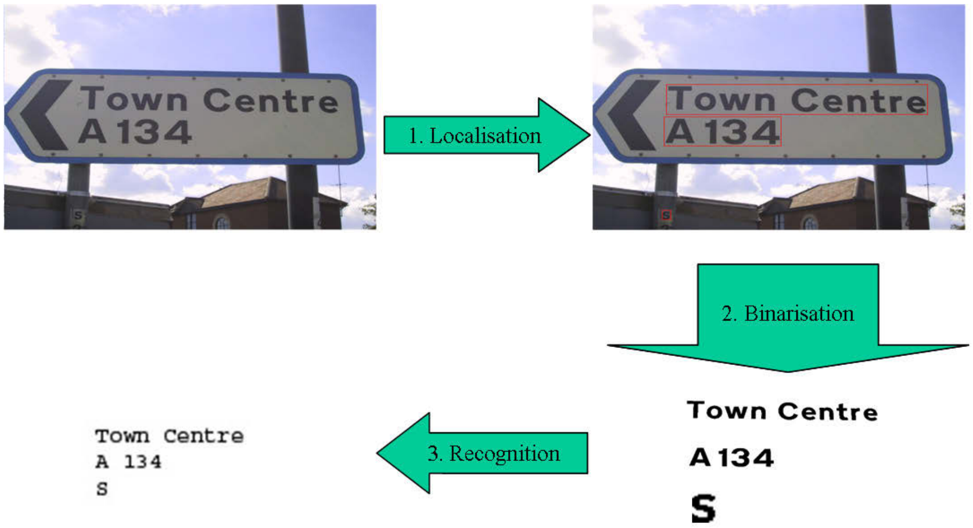


Figure 1.2: Three Steps of the TIE System.

components as illustrated in Figure 1.2: text detection, text binarisation and text recognition. Text detection is used to detect whether text contents appear in images, locate the positions of the text and find the areas of text using bounding boxes. Text binarisation is used to produce the binary version of the located sub-images containing text. Image enhancement is necessary when the extracted sub-images do not meet the requirement of quality before being sent for recognition. Text recognition is the process of turning the binarised and enhanced text contents into machine-editable text using an optical character recognition (OCR) system. As a summary, the final target of a TIE system is to automatically "read" text in images and video streams. Text information extraction (TIE) systems can be applied to many practical applications to convert text in acquired digital images into machine-editable text. Some TIE applications are listed as below.

- License plate recognition (LPR) [11]. This technology has been applied in intelligent transportation systems worldwide since the license plate is the exclusive identity of a vehicle. An intelligent transportation system embedded with LPR can be used for highway toll collection and city traffic analysis. It can also be used to retrieve stolen vehicles and enforce traffic rules.

- Content-based text information retrieval in multimedia. Content-based multimedia information retrieval (CBMIR) [12] refers to searching for information based on the actual contents of images or frames in video clips rather than keywords or tags. When CBMIR is applied to text information retrieval, the images having desired text information can be retrieved from database.

- Movie shot classification [13]. Motion is an important feature in the research of shot classification [14, 15]. If text appears in video frames, it becomes a disturbance and degrades the performance of shot classification algorithms. Text information can be detected by a TIE system and be removed by image restoration methods [16].

- Robot vision. As a component of robot's vision, the ability of reading text information in a real world environment is essential to guide the actions of a robot [17, 18].

- Support systems for visually impaired people. People with blindness or visual impairment can know how to use the home appliances by resorting to a TIE system [19] and "read" their instructions [20].

- Advanced driver assistance system (ADAS) [21]. Road sign recognition integrated in an ADAS can warn the driver when his/her vehicle exceeds the speed limit and can indicate overtaking restrictions.

1.2 Existing Methods of Text Detection and Binarisation

In this section, the existing framework text detection and for text binarisation are discussed with emphases on basic strategies. The detailed technical issues are discussed in Chapter 2.

1.2.1 Framework of Text Detection

Based on the initial assumptions of the text regions, there are two schemes as reported in the literature for text detection: the bottom-up scheme and the top-down scheme.

Bottom-up Scheme

This scheme considers a text region as a combination of multiple individual characters. Each character is a connected component [19,22–26]. Firstly, all of the possible character candidates are extracted by blob detection techniques [27], such as Maximally Stable Extremal Regions (MSER) [28]. Secondly, non-character connected components are filtered out by setting constraints on structure properties. A grouping step is performed to group the characters belonging to the same text line together to form text candidates. Finally, a text candidate verification step will be used to eliminate non-text regions. This scheme is suitable for images having high resolution since their text is clear and has high contrast against local background areas. The typical text samples of this kind can be seen in the dataset in ICDAR Competition of Reading Text in Scene Images [2,29,30].

Top-down Scheme

This scheme treats each text region as an integral area [31–37]. An image is first processed using image processing techniques, such as an edge detector to extract possible clues of text regions. Then, these clues are merged to form candidate text regions, which may be further refined if necessary. Lastly, the candidate text regions are verified by heuristic rules or trained classifiers. This scheme shows good performance for detecting texts in relatively low resolution images. Local text regions have obvious differences from the non-text regions in terms of edge distribution and intensity variance. Note that video text [31,33] detection algorithms mainly fall into this category.

1.2.2 Framework of Text Binarisation

The goal of text binarisation is to separate the pixels in a detected text image into foreground (text) and background. Edge information has been commonly used for text binarisation due to the inherent feature of text stroke. The edge contour can be filled in the inner side [10] or can be further explored to decide the text colour [38]. Based on the observation that the colour of characters in a text line is constant, clustering methods are applied to the colour text image and the cluster of text is left as the final binarisation result [9, 39]. As a special case of image binarisation, text binarisation also benefits from graph models that have been successfully used for image segmentation. Pixels are treated as nodes in a graph model at the first stage and a binarised version can be obtained following the graph-based image segmentation procedures [40].

1.3 Unsolved Problems

Although many text detection and binarisation algorithms have been reported, there are still several problems unsolved. When considering specific text detection and binarisation applications, there is not a single general algorithm that can be applied to all types of texts.

Typical characteristics of video text, digital-born text and natural scene text are basically different. For example, many existing video text detection methods can obtain very high precision rates and recall rates, however, they cannot be directly performed to detect natural scene text since the accuracy is much lower. This is the reason that different property assumptions are preset in the very beginning of creating a new method. If the corresponding assumptions cannot reveal the properties of the type of text, satisfactory performance cannot be achieved. The unsolved problems on text detection and text binarisation can be summarised into the following sub-problems:

Problem 1 for text detection: As text is the key research object in this research, exploiting the intrinsic characteristics of text in digitalised form is the main concern. These text characteristics should be discriminative from those of complex background near text regions. A high recall rate of text detection is sought.

Problem 2 for text detection: Apart from a high recall rate, a high precision rate is another factor to embody the overall effectiveness of a text detection algorithm. To what extent text differs from non-text is the most important issue in the fine detection stage. Although many features and heuristic rules have been studied to verify candidate text lines after coarse detection, more descriptive features for text should be investigated to filter out false alarms.

Problem 1 for text binarisation: The only purpose of text binarisation is to separate the pixels in a text image into two classes, one for text and one for background. Many local and global gray level thresholding methods can handle this ideally. Nevertheless, text images may be degraded by lighting and local complex background. More deliberate gray level image processing of text binarisation is necessary to overcome the degradation.

Problem 2 for text binarisation: Usually, the colour of text pixels in a text image is dominant compared with the colours in the background. How to make a good use of this observation and suppress the influence of the colour distribution of background is critical when binarising text from an image.

1.4 Research Objectives

In order to overcome the problems on text detection and text binarisation as listed in Section 1.3, the research work in this thesis aims to develop accurate and robust text detection and text binarisation algorithms. The text detection task is divided into two steps: the coarse detection and the fine detection. The objective of the coarse detection research is to increase the recall rate and the objective of the fine detection research is to improve the precision rate. In the text binarisation work, both gray level and colour information are considered. The proposed methods alleviate detrimental influence of lighting and local background to produce binarised text images with better quality.

1.5 Author's Contributions in This Thesis

By investigating the existing text information extraction systems, this thesis aims to develop effective text detection and text binarisation methods. The contributions of this thesis and the approach steps are outlined below.

- A multiple layer image scheme based on Maximum Gradient Difference (MGD) values is proposed for detecting text lines with low and high contrast.
- Two new texture-based features are adopted for text/non-text classification to eliminate non-text regions while keeping true text regions.

- A classifier based on geometry-based, stroke-based, HOG-based and colour-based features is trained for character/non-character MSER classification.
- Two strategies are proposed to retrieve the misclassified character MSERs to increase algorithm performance.
- A channel gray image selection scheme is developed to find the channel image having the biggest difference between foreground pixels and background pixels for text binarisation.
- A graph-cut method is applied to separate the selected channel image into foreground and background.
- A modified CC-based validation method is used to extract text areas after clustering.
- An objective segmentation evaluation method is proposed to determine the final text binarisation result from the two binarised text images.

1.6 Thesis Structure Overview

The thesis consists of six chapters. Chapter 1 presents the framework of text information extraction system and lists the existing problems in text detection and text binarisation. In Chapter 2, a comprehensive literature review on text detection and text binarisation is reported. Chapter 3 presents the proposed born-digital text detection method and demonstrate experimental results. In Chapter 4, an algorithm of natural scene text detection is given and is tested on a benchmark dataset. In Chapter 5, a text binarisation algorithm based on channel image selection and a CC-based text binarisation algorithm are presented and their effectiveness are shown. In Chapter 6, the work in this thesis is concluded and the future work is discussed.

Chapter 2

Review of Some Related Work

In Chapter 1, a brief introduction is given of existing methods for text detection and text binarisation. In this chapter, detailed information of the existing work for text detection and text binarisation is provided. The reviewed approaches are classified in accordance with different characteristics of text used in the literatures. Particularly, two manners of text/non-text regions classification for text detection are introduced.

Text binarisation algorithms classify the pixels of an image into two groups, namely, text pixels and background pixels and image segmentation algorithms partition an image into multiple pixel groups. In this sense, text binarisation is a special case of image segmentation. So, image segmentation techniques can be applied to text binarisation by reducing the number of classified pixel groups to two. Since graph-based approaches are quite effective methods for image segmentation, this thesis also explore the potential of graph-based image segmentation methods for solving the text binarisation problems. Hence, different types of graph-based image segmentation techniques are also discussed in the second half part of this chapter.

2.1 State-of-the-art Text Detection and Binarisation Methods

The focus of this section is to introduce the existing text detection and text binarisation methods. Text detection methods can be classified into edge-based, texture-based, colour-based and stroke-based methods. Text binarisation methods can be classified into edge-based, colour-based, stroke-based and graph-based methods.

2.1.1 Existing Text Detection Methods

A text detection algorithm usually consists of coarse detection and fine detection steps. Coarse detection is to conduct several basic image processing techniques to highlight text areas against background. Text candidate regions are the output of coarse detection step. These candidate regions usually contain too many false alarms. Therefore, a fine detection procedure is needed to remove those non-text candidates. Machine learning techniques and heuristic rules are commonly used for fine detection. According to the specific characteristics of text exploited in publications, text detection algorithms can be categorised into edge-based, texture-based, colour-based and stroke-based methods.

Edge-based Methods

Edge-based methods produce binary edge maps from input images first. Then, after the morphological dilation and erosion operations, connected components (CCs) are generated from the binary images. Prior knowledge of text regions is considered for further analysis in order to generate candidate regions. Similar methods have been utilised in the work of Jung, Liu & Kim [41] and Pan, Bui & Suen [42]. Classical edge detectors, such as Canny operator and Difference of Gaussian have also appeared in the literature. Jung, Liu & Kim [41] analysed each CC using horizontal and vertical axis projection profiles. Pan, Bui & Suen [42] labeled each connected component as text or non-text at pixel level and connected component level. A sparsity test using an over-complete dictionary, which was trained using the K-SVD algorithm, was the core of the labeling process. Shen et al. [43] grouped edges using a bottom-up hierarchy of more complex features. This approach stemmed from work on object-specific figure-ground segregation and was implemented using a "data-driven" graphical model. The top-level features (sticks and boxes) were used to construct the graphical model to detect the text regions.

Texture-based Methods

Texture-based methods are similar to edge-based methods. The main difference is that texture-based methods compute texture features of images first, rather than just producing the edges. Text regions usually have sharp changes of colour or intensity. In the work of Silapachote et al. [44], an image was firstly divided into square patches. Based on the assumption that text signs should belong to some generic class of textures, text signs were detected using local colour and texture features to classify image regions based on a discriminatively trained conditional maximum entropy model. In contrast to general texture-based methods, Zhu et al. [45] used the texture feature derived from text strokes. This detection method consists of three steps. They started with multiresolution processing through a single-level 2D Haar wavelet transformation. Then, thresholding and labeling were performed, which facilitated the utilisation of co-occurrence matrix to describe texture from strokes. Finally, the detection of candidate text region using texture features from strokes was done. A hybrid method of two heuristic algorithms was proposed by Kim et al. [46]

based on image intensity analysis. A Gray-level Information Analysis (GIA) method and Split and a Merge Analysis (SMA) method were proposed separately. The GIA system processed the input colour image by several preprocessing steps where image binarisation, and long line and noise removal were performed, before the extraction of candidate text region. In the SMA system, after the processes of split, merge and size restriction and dilation, the candidate text regions were extracted. The final text candidate regions were obtained by combining GIA and SMA.

Colour-based Method

By assuming that the colors of text and background in most of the road signs distribute homogeneously, the colour information grasps more and more attention. In the algorithm proposed by Ye et al. [47], the candidate text region was located by grouping text pixels through image segmentation and region layout analysis. A generalized learning vector quantization (GLVQ) algorithm was employed to group pixels of similar color into the same cluster in LUV colour space in image segmentation stage. After that, a spatial layouts analysis procedure was used to obtain candidate text lines. In Park and Park's method [48], clustering-based natural scene segmentation was firstly considered based on the histogram of hue and intensity components separately. Each candidate text region was normalised using nearest neighbor interpolation. The input wavelet feature vectors for the neural network were directly extracted from 64×64 normalised text regions. Kim and Kim [49] made use of colour information in another form. According to the observation that there existed transient colours between inserted text and its adjacent background due to colour bleeding, a transition map was produced which was utilised as an indicator for the overlay text region. Candidate text regions were extracted by a reshaping method and the overlay text regions were determined based on the occurrence of overlay text in each candidate.

Stroke-based Methods

Stroke is an intrinsic feature of text and its potential has been greatly explored in recent years. Local constraint and global constraint were used by Jung et al. [34] and Liu et al. [50] to define a sub-image of text. Local constraint was where many stroke-like structures were found in the sub-image, while global constraint was where these stroke-like structures had specific spatial distributions. A stroke filter was designed based on local region analysis, and spatial-similarity CCA was used to locate the text candidates. In the method of Subramanian et al. [51], the proposed system was a bottom-up approach and used a line-scan-based approach to find interesting areas having text-like properties. Two well-known features of text, approximately constant stroke width and local contrast, were exploited, and a fast, simple and effective algorithm was developed to detect character strokes for detecting text in further steps. In [25], Epshtein et al. proposed Stroke Width Transform (SWT) to employ the constant width of strokes in characters. A set of rules were utilised to group the characters with similar stroke width into a word. Character energy was defined in [52] based on the gradient direction of each stroke edge point and the distance between two mutually corresponding stroke edges. Character strokes normally have double edge of opposite gradient direction and this feature was explored in Liu et al.'s work [53]. Stroke-like edges were extracted to eliminate the impact of the non-stroke edges and a stroke-like edge operator was applied in a local neighbourhood of edges to seek possible parallel edges.

2.1.2 Region Classification

There are two cases of text/non-text region classification: coarse-to-fine framework and scanning window framework. In the first case, candidate text regions are generated after the image processing using edge, texture, colour or stroke features of text. Due to the unavailability of the general characteristics of background, many non-text regions may be deemed as candidate text regions. At this stage, a further refinement procedure is necessary to alleviate the non-text while keeping the text. In the second case, text regions are estimated by directly performing region classification in the original image.

Candidate Region-based Methods

After the input image is decomposed into a set of connected components or regions, the extraction problem is converted into a classification problem. Heuristics-based and classifier-based methods are two typical methods. In heuristics-based methods, whether a candidate region is kept as a text region or is filtered out as a non-text region is decided by empirical rules. In the method presented by Pan et al. [42], only those short “lines” having more than 80 percent of the edge points labeled as text were kept. Hanif et al. [54] used three empirical rules to verify the localised text regions as connected components. In the last step, all verified components were clustered together by applying the defined connected component rules. Classifier-based methods normally select certain features from candidate regions, and then feed these features into the trained classifier to do classification. In Jung et al.’s [41] approach, the classifier fusion of N-gray (normalised gray intensity) and CGV (constant gradient variance) were performed to verify text candidate, and then text line refinement based on the SVM output score, colour distribution and prior geometric knowledge was used to generate the final results. In the same way, wavelet coefficient histogram features and colour variance features were extracted to represent text and SVM was used as the classifier [47].

Scanning Window-based Methods

Different from the candidate region-based methods, the candidate regions are gained by exhaustive scan. These methods scan the whole image or a frame with a size-fixed window and predict the probability of being text of a region inside the window. Classifiers based on Boosting and SVM are applied to the vision related object detection systems.

Jung et al. [34] used a 15×15 sliding window to generate several samples. The SVM output scores of these samples are averaged. If the average is larger than a predefined threshold, the whole text line is regarded as a text line. Frome et al. [55] trained an SVM-based sliding window detector for localising license plates and human faces in large-scale images. A neural network-based post-processor was proposed to remove as many false positives as possible and keep almost all the true positives at the same time.

In recent years, the AdaBoost algorithm has been widely investigated and performed for text localisation task with selective choices of features. In Chen et al.'s work [56,57], statistical analysis was utilised to select the informative features of text to discriminate between text and non-text, and a cascade of 4 strong classifiers containing 79 features, which included statistics, gradient-related and edge features, was obtained after training. By using both global statistical features and local haar-like features, Zhang et al. [58] used global statistical features and local haar-like features together to make a classifier invariant to brightness, colour, size and position of the license plates. In Hanif et al.'s [54,59] method, a standard window was of 32×32 pixels or its integer multiples. Mean Difference Feature (MDF), Standard Deviation (SD) and Histogram of oriented Gradients (HOG) were extracted from text segments as three different types of features. A cascade AdaBoost classifier [60] was built up by connecting several AdaBoost strong classifiers in the text detection phase. The candidate feature pool was formed by using the histogram of oriented gradient (HOG) and multi-scale local binary pattern (msLBP) features. For text localisation, a window grouping method integrating text line competition analysis was used to generate text lines.

Apart from classifier-based method, clustering analysis [61] and transformation threshold [62] were also used to localise text regions. These two methods need to make transformation to original images first, and then wavelet features and DCT-based features were used for classification.

2.1.3 Existing Text Binarisation Methods

Before being sent to an OCR system, localised text regions need to be extracted from the image and converted into binary images. The processing results of text extraction highly affect the text recognition rate, so various methods on text extraction have been proposed under the efforts of researchers. Text binarisation can be edge-based, colour-based, stroke-based and graph-based methods.

Edge-based Methods

Edge-based methods usually make use of the edge information of the character strokes in a text in order to extract the text in subsequent processing steps. Kasar et al. [63] performed an edge-based connected component analysis and estimated the threshold for each edge connected component based on foreground and the background pixels. In [38], the union of edge information on R, G and B channel was used to generate an edge image. The representative colours in CIE L*a*b* space were obtained along the normal directions of the edge contour. These representative colours served as the initialisation of K-means clustering. In each colour cluster, connected component labelling was utilised and several constraints were defined to remove the non-text components. The final binarisation for each component was achieved by the threshold estimation similar to that in [63]. In the work of Yu et al. [64], an improved version of double-edge model based on the method in [65] was proposed to extract character strokes between the predefined minimum and maximum stroke widths. In [66], a stroke edge filter was devised to get the edge information of character strokes. The stroke edges were identified by a two-threshold scheme and the stroke colour was further estimated by inner pixels between edge pairs. The segmentation result was obtained by combining the binary stroke edges and strokes with four heuristic rules. Firstly, the edge of text was detected in [10] to get the text boundary. Secondly, the pixels inside the text boundary were selected with a low/high threshold as the seeds for the modified flood filling algorithm. Lastly, the false edges were removed by a morphological opening operation. However, the edge information could not be correctly achieved for the texts with uneven lighting or highlight, so the segmentation result was not good and it in turn resulted in recognition failure.

Colour-based Methods

Colour-based methods assume that the characters of text have the same colour information and are different from that of the background. In [67], the colour plane with the maximum breadth of histogram among Cyan/Magenta/Yellow colour space was chosen for adaptive single-character image segmentation. In [68], binarisation was performed on an optimally selected colour axis, on which it had the largest inter-class separability in the RGB colour space. Thillou et al. [69] pointed out that the selection of clustering distances was the main problem of the degraded natural text segmentation after investigating several colour spaces for clustering. With the combination of Euclidean Distance and Cosine Similarity [9], the pixels of each natural scene text image were clustered into three categories: textual foreground, background and noise. The final segmentation result was achieved after the implementation of Log-Gabor filters. Song et al. [70] performed colour reduction and clustering using the Euclidean distance to reduce the number of colours and processing time. Each colour cluster was treated as a colour plane, and each colour plane was classified into text, background, noise or text edges. In the work of Mancas-Thillou et al. [71], the Euclidean distance and the Cosine similarity were used for K-mean clustering. Log-Gabor filters were chosen to combine colour and spatial information to segment characters properly. Based on the observation that the hue value of chromatic pixels changed less than the lightness value under shadows and uneven lighting condition, Yao et al. [72] classified text regions into three types: gray text region, chromatic text region, gray and chromatic mixture text region. Segmentation algorithms based on hue, hue and lightness, and lightness were performed respectively for the three types of text regions respectively. Colour-based methods describe the chromaticity differences between text and background which can be used for segmentation.

Stroke-based Methods

Stroke-based methods search the stroke-like structures in text images using a stroke filter. Considering strokes as intrinsic text characteristics in [73, 74], a stroke filter was designed to get the response of text strokes. After the text colour polarity was determined, a local region growing procedure was performed to refine the binarised stroke response map. This approach mainly makes use of the transitional colour between the strokes and the adjacent background in embedded video text images. Unfortunately, this approach may not work for scene text as there is usually no transitional colour between scene text characters and the adjacent background.

Graph-based Methods

Graph theory is such an active theory that it is commonly used in different research fields and text segmentation is not an exception either. Li et al. [75] utilised graph theory to handle multi-polarity (multi colours or intensities in the same line) text segmentation. In this model, a colour image was firstly converted into its intensity map, and then it was represented with an undirected weighted graph by treating pixel groups at various gray levels as nodes. Weights of edges linking these nodes were defined as correlations of these pixel groups. Each intensity map could be effectively split into several single-polarity text images with continuous gray levels. These single-polarity images with texts were selected by a trained SVM classifier. After post-processing stage using seed fill algorithm, hue histogram analysis and connected components analysis, the binary text image of a colour image was generated. In Mishra et al.'s work [40], the Markov Random Field (MRF) based graph model was constructed for binarising natural scene text images. Image pixels were represented as nodes in a MRF and a new energy function was introduced. A Gaussian Mixture Model was used in this energy function. The energy function was minimised by an iterative graph cut strategy to find the optimal binarisation.

2.2 Recent Advances on Graph-based Image Segmentation Techniques

Image segmentation techniques using graph theory have become a thriving research area in computer vision community in recent years. This chapter mainly focuses on the most up-to-date research achievements in graph-based image segmentation published in top journals and conferences in computer vision community. The representative graph-based image segmentation methods included in this chapter are classified into five categories: minimum-cut/maximum-flow model (called graph-cut in some literatures), random walk model, minimum spanning tree model, normalised cut model and isoperimetric graph partitioning. The basic rationales of these models are presented and the image segmentation methods based on these graph-based models are discussed as the main concern of this section. Several performance evaluation methods for image segmentation are given. Some public databases for testing image segmentation algorithms are introduced and the future work on graph-based image segmentation is discussed at the end of this section.

2.2.1 Introduction to Image Segmentation

In computer vision applications, image segmentation is to separate an image into several regions, of which each has certain properties in terms of predefined rules. These regions can represent objects, parts of an object, or background. The aim of image segmentation is to find the regions of interest (ROI) in an image according to a particular application.

As a very important technique in computer vision, image segmentation has been found in a wide variety of practical applications, such as medical image processing, satellite image analysis, biometric recognition (e.g. human face recognition, fingerprint recognition and palm recognition), traffic control systems (e.g. vehicle number plate recognition, vehicle counting), digital photo editing, robotic vision and so on. Image segmentation approaches include clustering methods [76], compression-based methods [77], histogram-based methods [78], edge detection methods [79], region growing methods [80], partial differential equation-based methods [81], graph-based methods [82], watershed transformation methods [83] and so on. Among these image segmentation methods, the graph-based approach has attracted many attentions and become one of the most thriving research areas in computer vision in recent years. Meanwhile, many high quality papers on graph-based image segmentation techniques have been published in top journals and conferences in the field of image processing and computer vision. This chapter intends to gather the information on the most up-to-date graph-based image segmentation research together and gives a clear overview of research work on this topic.

The remaining parts of this section are arranged as follows. Firstly, mathematical knowledge of graph theory is given. We then present the five most representative graph-based models used for image segmentation and classify them into two major groups based on whether interactions of users are involved. Then, methods used for evaluating segmentation performance are discussed. Some benchmark image segmentation datasets are introduced. Conclusions are given in the end.

2.2.2 Background

Mathematically, a graph $G = (V, E)$ is composed of a set of nodes (or vertices) V and a set of edges E. An edge $e \in E$ is the connection of two nodes $v_i \in V$ and $v_j \in V$. There are two types of edges: undirected edge and directed edge. An undirected edge is an unordered pair of nodes $e_{\{v_i,v_j\}}$. A directed edge is an ordered pair of nodes $e_{\{v_i,v_j\}}$, in which v_i is

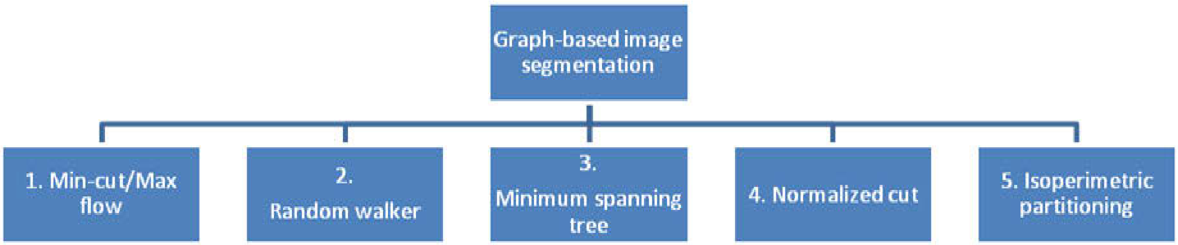


Figure 2.1: The categorization of graph-based image segmentation techniques

called the starting node and v_j is called the ending node. The weight of an edge is a value assigned to the edge, which describes the relationship between the two nodes.

In graph-based image segmentation models, a node can be a pixel, a set of pixels with common characteristics, a super pixel, or a feature vector of the pixel. The edge weight w is defined to describe the similarity or dissimilarity of two nodes connected by it according to the specific application.

The categorisation of graph-based image segmentation techniques can be seen from Figure 2.1. Following this structure, the image segmentation techniques based on the five models will be discussed one by one. The relative graph theory knowledge will be presented first followed with the detailed content of different techniques. Meanwhile, the above mathematical notations of graph, node and edge are used in the following content.

2.2.3 Supervised Graph-Based Image Segmentation Methods

Supervised segmentation methods involve the interactions of users, such as marking some pixels for object and background or drawing a rectangle which embracing the whole object to be segmented. This kind of segmentation methods can obtain the desired foreground and background. Among the five reviewed models, the minimum-cut/maximum flow model and the random walk model are used for supervised image segmentation.

Minimum cut/maximum-flow Models

In minimum-cut/maximum flow (also called graph cut) image segmentation model, an image is represented by an undirected graph $G = (V, E)$. Two special additional terminals are included in the graph, and they are usually called the source (object terminal), denoted by s, and the sink (background terminal), denoted by t, as shown in Figure 2.2. Let P and N denote a set of pixels in an image and a set of all unordered pairs $\{v_1, v_2\}$ of neighboring pixels in P respectively. Let $A = (A_1, \cdots, A_p, \cdots, A_{|p|})$ denote a binary vector of which each component is a label A_p assigned to a pixel p representing either object ("obj" in abbreviation) or background ("bkg" in abbreviation). The binary vector A records a segmentation result. Among all of the possible segmentations, the optimal segmentation is found by incorporating soft constraints and hard constraints. The soft constraints are the boundary and regional properties of segmentation A described by the cost function

$$E(A) = \lambda \cdot R(A) + B(A) \tag{2.2.1}$$

where

$$R(A) = \sum_{p \in P} R_p(A_p) \tag{2.2.2}$$

$$B(A) = \sum_{(p,q) \in N} B_{(p,q)} \cdot \delta(A_p, A_q) \tag{2.2.3}$$

and

$$\delta(A_p, A_q) = \begin{cases} 1 & if A_p \neq A_q \\ 0 & otherwise. \end{cases} \tag{2.2.4}$$

In 2.2.1, $R(A)$ and $B(A)$ are usually called the regional term (or data term) and the boundary term (or smoothness term) respectively. λ is a non-negative scalar to adjust the trade-off of importance between the regional and boundary terms. In 2.2.2, $R_p(A_p)$ denotes the penalty for assigning A_p to pixel p. In 2.2.3, $B_{\{p,q\}}$ denotes the penalty for discontinuity between p and q. The hard constraints are some restrictions imposed by user interaction. In interactive segmentation method, one popular constraint is that some pixels are marked as “obj”and some pixels are marked as “bkg”by a user:

$$\forall p \in O, A_p = \text{“}obj'', \forall p \in B, A_p = \text{“}bkg''. \tag{2.2.5}$$

where O and B represent the sets of pixels marked as “obj”and “bkg”respectively. At the meantime, $O \subset P$ and $B \subset P$ such that $O \cap B = \emptyset$.

Following the hard constraints, the remaining unmarked pixels are assigned to “obj” or “bkg” during the optimal segmentation $A_{optimal}$. $A_{optimal}$ globally minimises the cost function in 2.2.1 among all of the possible “obj/bkg” segmentations of the given image. The minimum cut $C_{optimal}$ is a subset of E which bi-partitions the graph corresponding to $A_{optimal}$. As a theoretical support, the existence of an optimal segmentation defined by the minimum cut that minimizes the cost function 2.2.1 among all segmentations satisfying the hard constraints in 2.2.5 has been proven in [82]. A concise presentation of the min-cut/max-flow model is shown in Figure 2.2. Figure 2.3 gives a segmentation example based on the graph model constructed in Figure 2.2.

Based on the original min-cut/max-flow algorithm, some researches focus on modification for the purpose of obtaining improved algorithms with better segmentation performances. In [3], a new min-cut/max-flow algorithm was developed and its efficiency

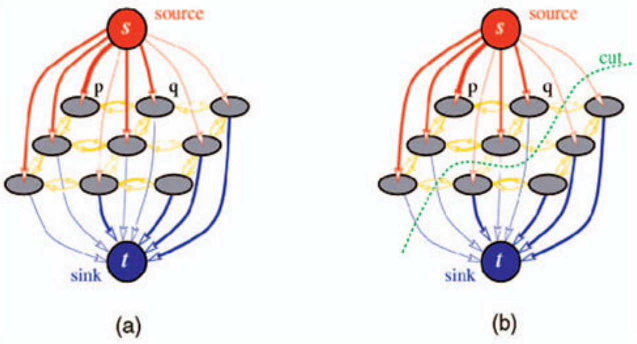


Figure 2.2: (a) The illustration of constructed min-cut/max-flow model. (b) A cut on the constructed graph. (courtesy of [3]).

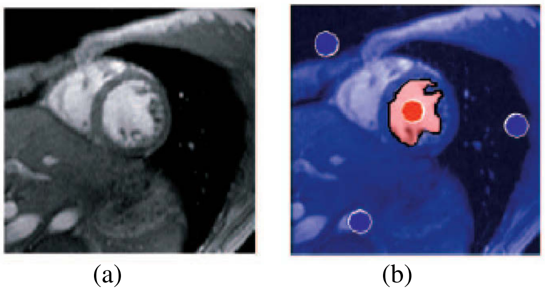


Figure 2.3: (a) Original image. (b) The segmentation result of minimum cut. (courtesy of [3]).

was compared with three standard algorithms belonging to Goldbery-Tarjan style “push-relabel” methods and Ford-Fulkerson style “augmenting paths”. Based on augmenting paths, the new algorithm built two search trees starting from the source and the sink. The algorithm iteratively repeated the “growth” stage, the “augmentation” stage and the ”adoption” stage, and it terminated when the two search trees could no longer grow and the trees were separated by saturated edges. The termination condition represented that a maximum flow was found which meant the minimum cut was achieved. Although the complexity could be worse than the standard algorithms in theory, the proposed algorithm greatly outperformed standard algorithms on typical problem instances in vision (including interactive image segmentation) in terms of running time based on the experimental comparison.

Yuan [84] presented a study on the max-flow models in terms of the continuous case instead of the discrete situation. The proposed continuous max-flow model corresponds to the continuous min-cut formulation as in the case of normal discrete situation. New explanations of the basic conceptions were given to describe graph cuts in continuous model. Based on the continuous max-flow formulation, new interactive (stated as supervised) graph-cut algorithms were developed and the complexities were the same as the unsupervised ones.

The selection of the parameter λ in the cost function is not trivial, because different choice of λ can highly affect the segmentation result. Efforts have been made to get an optimal λ value. Peng et al. [85] observed that different values of parameter λ might result in over-segmentations, good segmentations and under-segmentations, so a supervised algorithm for automatic parameter selection was proposed to obtain the best segmentation for each image. A measure of segmentation quality was developed based on intensity, gradient, contour continuity and texture features which were normalized by a novel way. After performing graph cut segmentation for each λ, the segmentation with the highest segmentation quality was chosen according to the measurement learnt by AdaBoost algorithm. In [86], according to the observation that the same parameter λ in the cost function would not be a good choice for the whole image, a method adaptively changing λ for different regions of an image was proposed to overcome over-segmentation and under-segmentation problems. Canny edge detection algorithm was performed on the given image at different hysteresis threshold. Then, the edge probability of each pixel $\overline{I}_i$ was calculated based on the generated edge maps. By changing λ into $(1 - \overline{I}_i)\lambda$, the effect of the regional term and the boundary term became adaptive for different regions of the image.

Combining two approaches to exert the advantages of each other to generate a more effective method is a common idea in practical issues. Working together with other models, the min-cut/max-flow model could produce better segmentation results. Borrowing the ideas of the topology preserving level set method in [87], a novel min-cut/max-flow algorithm embedding geometric prior knowledge constraints was developed for image segmentation in [4]. Five elements (a foreground/background label attribute, a level set style initialization, inter-label and intra-label stages of max-flow computation, a distance map and a bucket priority queue data structure) were considered to implement the discrete graph-based algorithm. Topology was preserved by checking the simple point [88] condition. The proposed algorithm could get better medical image segmentation results than the graph cut method (as shown in Figure 2.4).

In [89], both the geodesic approach and the graph cut approach were revisited. Moreover, the causes of their segmentation errors were analysed. The geodesic segmentation approach and the graph cut method were combined for interactive image segmentation task. The cost function in the min-cut/max-flow graph cut framework was modified by

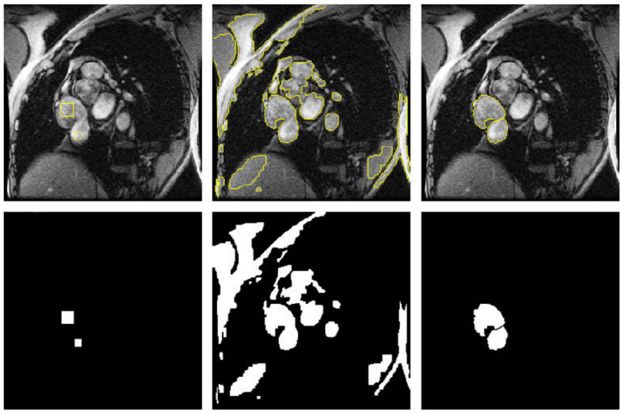


Figure 2.4: (a) Initialisation. (b) Segmentation result of minimum cut. (c) Segmentation result of topology cut (courtesy of [4]).

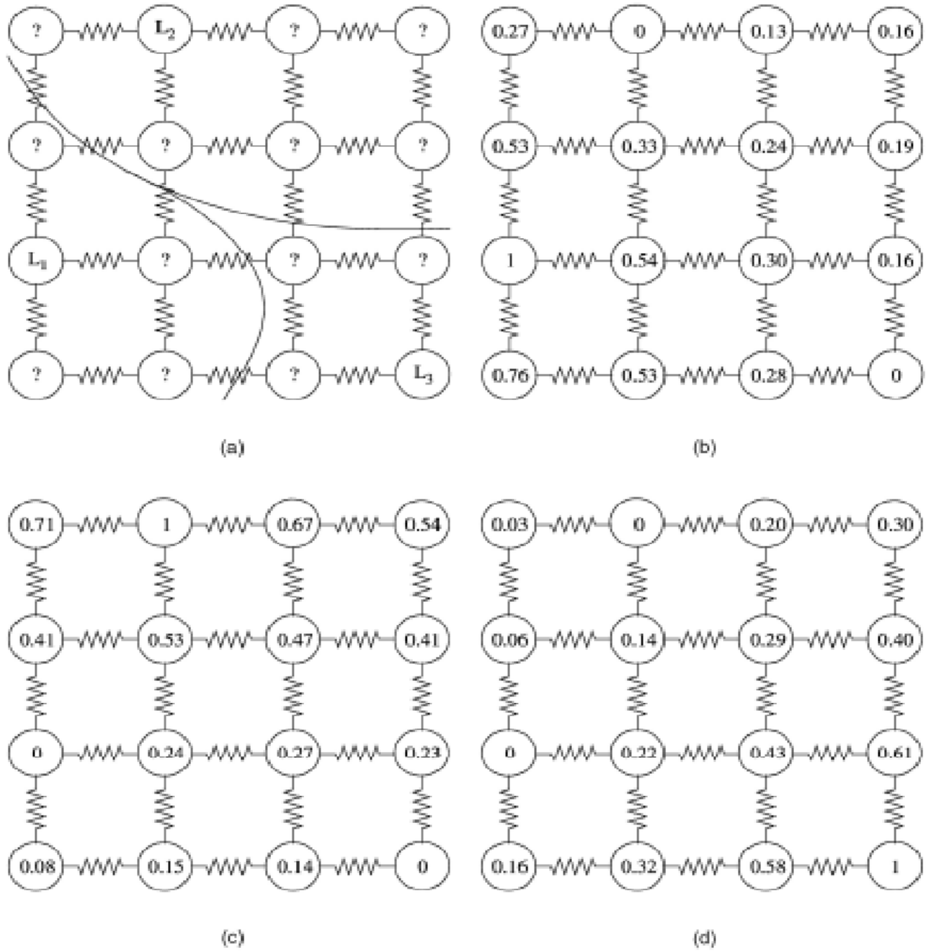
adding geodesic distance information into the regional term. Furthermore, the boundary term was adjusted to incorporate geodesic confidence. The proposed method was superior in two aspects: less sensitive to shortcutting than standard graph cut methods and less prone to seed placement and better at edge localization than pure geodesic methods.

Tensor voting framework was embedded into graph cuts using principles of perceptual grouping [90]. The tensor map was generated by adding the votes at each receiving site. A local Riemannian metrics was designed to encode tensor voting framework into the weight of edge in the graph construction. Compared with isotropic metrics, better results were obtained by the proposed framework as both the edge strength and the edge orientation were taken into account. Compared with the flux-based approach, the proposed framework did not need shape priors.

In some applications, the user's interaction is infeasible for segmentation task, so the full automation of image segmentation using a graph cut model has become a common concern. Usually, an automatic pre-processing functioning as a user interference is indispensible. Due to the observations that user interactions could lead to a wrong segmentation or not be feasible in some cases, Fu et al. [91] presented an automatic object segmentation method called Saliency Cuts. The saliency detection was performed firstly. Then, the "Professional Labels" were generated automatically by the multi-resolution framework as the imposed seeds for the object and the background. Finally, the graph cuts algorithm was implemented for the segmentation. The experimental results showed the advantages of automation and accuracy of the proposed method. Jung et al. [92] proposed a novel and fully automatic scheme for segmenting objects from images. A saliency detection method was used as an automatic generation of object seeds and background seeds. As the generated object seeds may position at the parts belonging to background or vice versa, an iterative self-adaptive graph cut method was developed to reduce the wrongly positioned seeds.

Random Walk Models

The random walker algorithm for multilabel and interactive image segmentation was described in [5, 93] for the first time. Based on its theory, the definition of a "walk" was a sequence of nodes and edges, both starting and ending with a node, and there was no restriction on the number of times a node can be visited. An edge weight of the graph was treated as a probability in the random walker algorithm. The probability of an unlabeled pixel reached one of the seeds with a pre-marked label by a random walk was calculated. If an unlabeled pixel has a higher probability to reach one seed with a label L than other seeds with other labels, this pixel was assigned to label L. In this way, the segmentation was accomplished by assigning all of the unlabeled pixels to one of the pre-marked labels. An illustration of the random walker algorithm is shown in Figure 2.5. The random walker algorithm presented in [5] is of the following qualities: fast computation, fast editing, an



(a)

(b)

(c)

(d)

Figure 2.5: Illustration of the random walker algorithm. (a) Initialisation of seed points L1, L2, and L3. (b) Probability of a random walker starting from each node first reaches L1. (c) Probability of a random walker starting from each node first reaches L2. (d) Probability of a random walker starting from each node first reaches L3. (courtesy of [5]).

ability to produce an arbitrary segmentation with enough interaction and intuitive segmentations.

In [94], a general framework of interactive image segmentation was presented. The min-cut/max-flow algorithm and the random walker algorithm corresponded to the cases of ℓ_1 norm and ℓ_2 norm of energy minimization respectively. A new segmentation algorithm based on ℓ_∞ norm was proposed which was more stable in terms of the seed number.

How to increase the processing speed of image segmentation algorithms is always a pursuit of researchers. In order to speed up image segmentation, an offline precomputation was performed before marking the seeds for object and background by a user [95]. A linear-time approximation of the random walker algorithm was developed by using several eigenvectors of the weighted Laplacian matrix of a graph. The proposed algorithm could converge to the random walker segmentation using more eigenvectors. With the offline precomputation of the segmentation, the final segmentation could be generated around 14 times faster than the original random walker algorithm [5] on MATLAB. Although the prior knowledge can make the segmentation more accurate [96], not all the prior knowledge is available before the user interaction. Similar to [95], a method for speeding up the medical image segmentation was proposed in [97]. An algorithm combining the random walker with priors and additional offline precomputation was derived to increase the speed of segmentation process.

For the purpose of making interactive image segmentation algorithms understand more intelligently the seeds provided by users, Yang [98] created a random walk-based algorithm which could make use of three types of seeds: the seeds for object and background, the seeds indicating the region that the object boundary should pass through, and the seeds specifying the pixels that the boundary must align with. The last two kinds of seeds, which were different from the normal object or background seeds, required local editing to refine the segmentation.

2.2.4 Unsupervised Graph-Based Image Segmentation Methods

Unsupervised segmentation methods divide an image into several regions based on some predefined metrics without the interference of users. The minimum spanning tree-based method, the normalised cut method and the isoperimetric method are deemed as unsupervised image segmentation.

Minimum Spanning Tree Model (Local Variation)

In [6], a segmentation of an image was defined as a partition of all nodes into components such that each component belonging to the segmentation corresponds to a connected component. A predicate D was defined to describe the existence of a boundary between two components. The internal difference of a component was defined as the largest weight in the minimum spanning tree of the component. The difference between two components was defined as the minimum weight of the edges connecting the two components. If the difference between two components was greater than the minimum internal difference of the two components, it was judged that there was a boundary between these two components. Otherwise, there was no boundary between these two components. As this algorithm considered local properties, it was also called local variation-based image segmentation in the literature. An intuitive example of minimum spanning tree-based method was presented in Figure 2.6. Following the predicate D, the proposed image segmentation was neither “too coarse” nor “too fine”, and this property was proved.

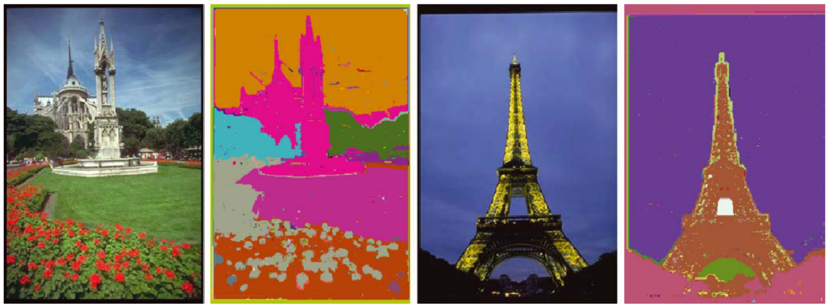


Figure 2.6: Results of minimum spanning tree-based image segmentation (courtesy of [6]).

Normalised Cut (Spectral Graph Theory-based)

In [99], the image segmentation problem was transferred to a graph partitioning problem. To avoid cutting small sets of nodes as what the minimum cuts do [100], the normalised cut, denoted as $Ncut$, was proposed as a global criterion to evaluate the disparity between different groups and the homogeneity inside the groups. The normalised cut of two disjoint sets A and B in a graph $G = (V, E)$ was defined as below:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}. \tag{2.2.6}$$

where $cut(A, B)$, $assoc(A, V)$ and $assoc(B, V)$ were defined as those in Equations 2.2.7-2.2.9.

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v), \tag{2.2.7}$$

$$assoc(A, V) = \sum_{u \in A, t \in V} w(u, t), \tag{2.2.8}$$

$$assoc(B, V) = \sum_{s \in B, t \in V} w(s, t). \tag{2.2.9}$$

The minimisation of the normalised cut was transferred into solving an eigenvalue system. The eigenvector with the second smallest eigenvalue of the eigenvalue system was used to bipartition the graph. The final segmentation was obtained by recursive repartition of the segmented parts.

In [101], the normalised cut algorithm and prior knowledge constraints were combined together to enhance the performance of image segmentation. At this point, the proposed method was analogical to interactive min-cut/max-flow method. As the constraints could not be incorporated with the original normalized cut method, an iterative approach was used and its convergence was guaranteed. The proposed algorithm could also be a solution to other graph cut methods.

Turbo pixels were used as the nodes in a graph and the normalised cut algorithm was performed on the graph to get the final image segmentation in [102]. The original input image was over-segmented into turbo pixels which were enforced to be convex. In this way, the image segmentation process consumed less time, while good segmentation results were still available.

In [103], the first polynomial time algorithms were developed to solve the ratio region problem and the variant of normalised cut problem. The normalised cut variant problem belongs to the optimisation of the ratio of the similarity within each group over the dissimilarity between groups, which is the target of image segmentation.

Isoperimetric Graph Partitioning

Following the classic isoperimetric problem in geometry (that was to find the region with minimum perimeter for a fixed area), [5] proposed an isoperimetric algorithm for image segmentation by dividing an image into regions with large areas and small perimeters. For a graph $G = (V, E)$, the isoperimetric ratio of an isoperimetric set $S \subset G$ was defined as

$$h(S) = \min_{S} \frac{|\partial S|}{Vol_S}. \tag{2.2.10}$$

In 2.2.10, the boundary of S is defined as $\partial S = \{e_{\{v_i,v_j\}} | v_i \in S, v_j \in \overline{S}\}$ and $\overline{S}$ is the complementary set of S. $|\partial S| = \sum\limits_{e_{ij} \in \partial S} w(e_{ij})$ and $Vol_S = \sum\limits_{i} d_i \forall v_i \in S$. The target of the isoperimetric algorithm is to maximize Vol_S and minimize $|\partial S|$. In contrast with the normalized cut algorithm, isoperimetric graph partitioning is faster and more stable.

2.3 Summary

In this chapter, we review the existing techniques on text detection and text binarisation. Text detection methods are categorised into edge-based, texture-based, colour-based and stroke-based methods in terms of the different properties embody in the text regions that can be taken to discriminate from non-text regions. Candidate region-based and scanning window-based approaches are also surveyed as they are mainstream frameworks for text/non-text classification. Text binarisation techniques which convert the detected text regions into binary images facilitate the recognition of the detected text. Edge-based, colour-based, stroke-based and graph-based text binarisation methods are presented.

As what is claimed in the second paragraph of this chapter, text binarisation is a special case of image segmentation and graph-based approaches have demonstrated their effectiveness for image segmentation problems. Therefore, graph-based scheme is considered for the text binarisation problem. Recent advances on graph-based image segmentation techniques are studied. The state-of-the-art graph-based image segmentation techniques have been reviewed. Those techniques have been classified into five models (min-cut/max-flow model, random walk model, minimum spanning tree model, normalised cut model and isoperimetric graph partitioning model) according to the specific graph models in the algorithms. Those techniques have been published in quality international conferences and international journals in computer vision research area and can be regarded as a trend of research in image segmentation. Due to the fact that the assessment of the performance of image segmentation methods is still an open question, many researchers are making efforts in providing supervised and unsupervised evaluation methods. As unsupervised evaluation methods can be included into the framework of image segmentation, they are more suitable to real world applications than the supervised methods.

Chapter 3

Born-digital Text Detection

In the comprehensive structure of the text information extraction system illustrated in Chapter 1, text detection is the preliminary step. The successive steps heavily depend on the quality of the result of text detection. A good text detection algorithm can accurately locate the text regions as well as effectively eliminate the noise from the background.

In this chapter, a new bottom-up text detection framework consisting of coarse detection and fine detection is presented. The novelty of the proposed framework is that a multiple layer image scheme is used for detecting text lines with strong and weak edge strengths. The maximum gradient difference (MGD) [36] is further developed to better describe the edge strength of an image for finding the possible text lines. The edge information of text regions is extracted based on clustering MGD values. Multiple layer images are generated for detecting text lines with strong and weak MGD responses. In order to distinguish text from non-text regions in the fine detection, a variant of local binary pattern (LBP), namely T-LBP, is proposed to depict the characteristics of text in the viewpoint of treating text as a kind of texture and use a supervised learning scheme to remove the false alarms generated in the coarse detection step. By further improving the idea of T-LBP and considering not only the horizontal and vertical directions of character strokes but also their diagonal and anti-diagonal directions, another new variant of LBP-based feature descriptor, namely IT-LBP, is also proposed in this chapter. The flow chart of the proposed method is illustrated in Figure 3.1.

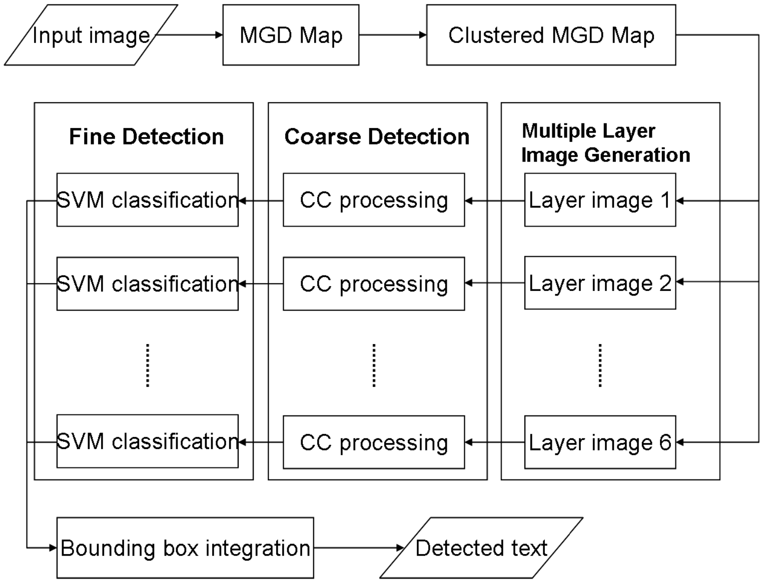


Figure 3.1: The flow chart of the proposed method.

The structure of this chapter is organised as follows. The coarse detection is introduced in Section 3.1 and the fine detection is discussed in Section 3.2. Different low-level features of text regions are discussed in Subsection 3.1.1. Secondly, Subsection 3.1.2 discusses about multiple layer images generation. Morphological operations for processing the binary clusters are presented in Subsection 3.1.3. Then, post-processing on the connected components is presented in Subsection 3.1.4. In Subsection 3.2.1, the features for text verification is presented. Supervised machine learning methods are shown in Subsection 3.2.3. Bounding box precessing is analysed in Subsection 3.2.4. The comparisons between the proposed method and other text detection algorithms are given in Section 3.3. A summary is given in Section 5.4.

3.1 Coarse Detection

The final target of a text detection algorithm can be split into two basic goals: accurately locate the text lines and suppress the non-text noises from the background. In our framework, coarse detection plays the role of locating the regions of the text lines. However, some non-text regions having similar structures of the text remain in the regions. Therefore, a fine detection step is necessary to eliminate the false alarms. In this section, the technical details of coarse detection are presented.

3.1.1 Maximum Gradient Difference

To start with, the instinctive characteristic of text lines should be explored. A text line is the alignment of characters and each character is formed by strokes. The high frequency of stroke-background transition in the text lines can be easily observed with a careful attention. This stroke-background transition provides a clue of the existence of text. The transition implies the sharp changes of intensity in the text lines. Those points having sharp change in brightness form the edge of the text lines. Here, we conclude that the text lines have high density of edge points. Taking the advantage of this property, many text detection methods have been proposed based on edge information. Edge detection consists of smoothing, enhancement, detection and localisation. The common edge detectors are Sobel Operator, Prewitt Operator, Roberts' Operator and Canny edge detector. Different edge detectors are based on different masks for convolution. The Sobel edge mask is illustrated in Figure 3.2 as an example.

Among those edge detectors, Canny edge detector [104] has been widely used due to its advantages of good detection, good localisation and minimal response. However, the setting of the thresholds in the Canny edge detector can affect the effectiveness. If the

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \qquad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I$$

Figure 3.2: The computation equations of G_x and G_y. I denotes the source image, G_x and G_y are the horizontal and vertical derivative approximations. $*$ represents the convolution operation.

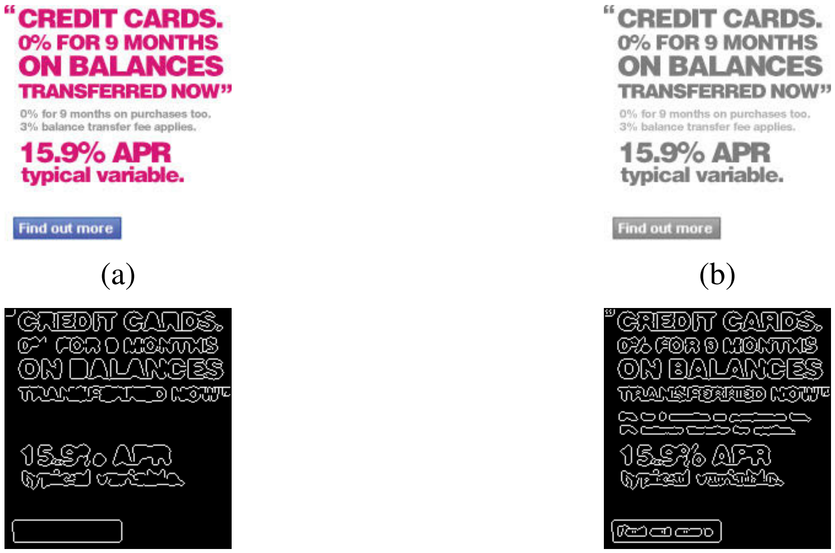
thresholds are set too high, some important edges may not be detected. Likewise, too many redundant edges may be extracted with too low thresholds. In the case of extracting the edges of text with low contrasts may not be successfully extracted when the thresholds are set too high. Since the result of Canny edge detection of an image is a binary map, the lost edge information can not be regained in the later processes. Although changing the thresholds may retrieve the complete edges of a certain image, the same threshold setting may not be suitable for other images.

Based on the above discussion on edge information extraction, using binary edge information as the basic hints for detecting text has negative effects. Comparing with binary edge information, the magnitude of gradient provides information about the strength of the edge rather than the existence of the edge. The gradient information is not filtered out by thresholding and can reflect the contrasts of all text regions. The gradient of an image f is a vector with its magnitude and direction given below:

$$\text{Gradient: } \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right).$$

$$\text{Gradient magnitude: } \|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}.$$

$$\text{Gradient direction: } \tan^{-1}\left(\frac{\partial f}{\partial y}/\frac{\partial f}{\partial x}\right).$$



(a) (b)

(c) (d)

Figure 3.3: An example of Canny edge maps obtained with different thresholds. (a) The original colour image. (b) The gray-level image of (a). (c) The Canny edge map of (b) with high thresholds. (d) The Canny edge map of (b) with low thresholds.

The gradient can be calculated by using a finite difference approximation as given below:

$$\frac{\partial f}{\partial x} = \frac{f(x+h_x, y) - f(x,y)}{h_x} = f(x+1, y) - f(x,y).$$
$$\frac{\partial f}{\partial y} = \frac{f(x, y+h_y) - f(x,y)}{h_y} = f(x, y+1) - f(x,y).$$

The changes of intensity incur the transitions between strokes and the local background in the text regions. In other words, there is a high frequency of transformation between the low gradient values and the high gradient values. Maximum Gradient Difference (MGD) [36] is the difference between the maximum and minimum gradient values within a local window of size $1 \times n$ centered at a pixel. Text regions usually have larger MGD values than non-text regions no matter if they are the dark texts on light background or light texts on dark background. This property has been adopted in several existing work on text detection [31, 36, 105, 106]. The common way of separating the pixels of potential text regions from the pixels belonging to background is the binary classification of the image pixels according the MGD value of each pixel. This means that the pixels with greater MGD values are classified as text pixels and the pixels with lower MGD values are classified as non-text pixels. This purpose can be completed by using k-means clustering where $k = 2$ as in [31].

However, it is abrupt to simply bi-separate all of the pixels in a image into "text" group and "non-text" group. MGD values of different text regions may vary in a wide range. As a result, the pixels of text lines with low MGD values could be erroneously clustered as non-text pixels. In order to avoid missing text regions with low contrast, multiple MGD-based clustering by setting $k = 4$ in the k-means algorithm is used. The horizontal gradient map g of the gray scale image is obtained due to the richness of vertical strokes of texts. The MGD value for the pixel (x, y) is then computed as the difference between the largest and the lowest gradient values in a 1×21 horizontal neighbourhood window as shown below.

$$MGD(x, y) = max(g(x, y - t)) - min(g(x, y - t)), t \in [-10, 10]. \tag{3.1.1}$$

3.1.2 Multiple Layer Image Generation

In the present approach, the pixels of the MGD map are classified into four clusters known as CC cluster maps which are denoted as $CCMAP_i(i = 1, 2, 3, 4)$. The order is sorted according to the means of the four clusters from smallest to greatest. One example is shown in Figure 3.13. In Figure 3.13(b), the four colours represent the four clusters obtained by k-means. It can be clearly seen that the pixels with different MGD responses belong to different clusters. The four clusters are represented by four colours ($CCMAP_1$ is red, $CCMAP_2$ is green, $CCMAP_3$ is blue and $CCMAP_4$ is white).

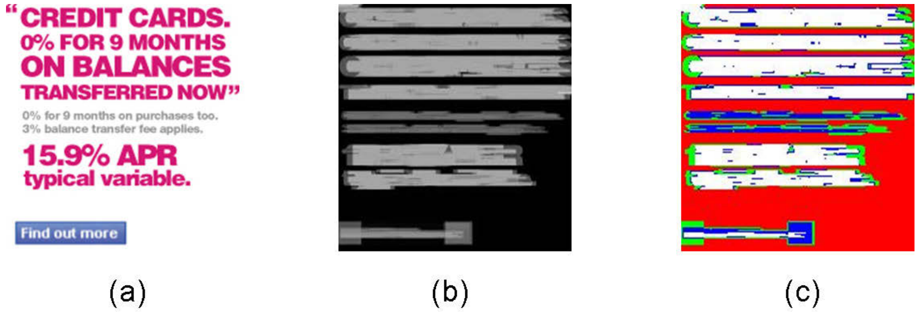
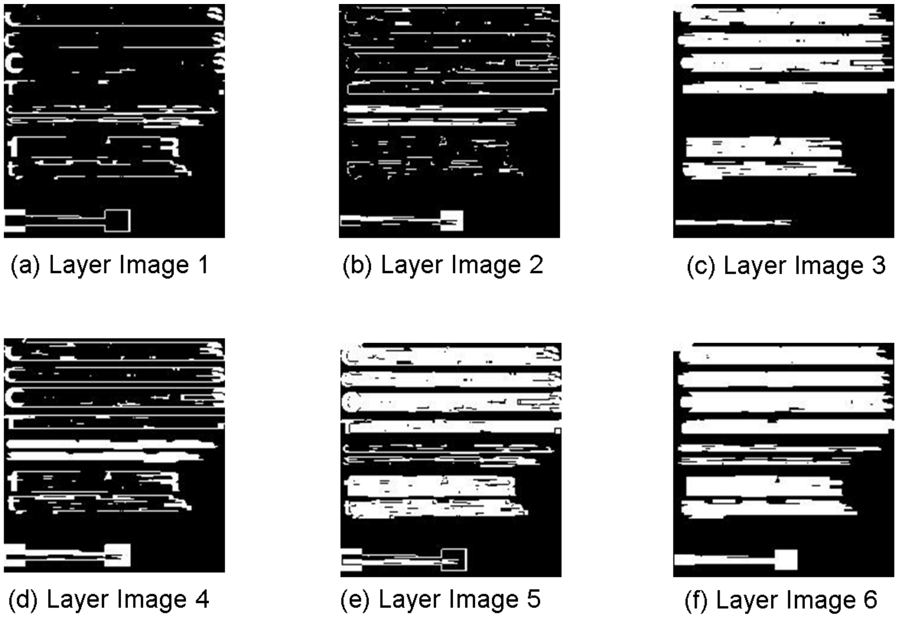


Figure 3.4: An example of MGD map clustering. (a) An original image. (b) MGD map of (a). (c) the four clusters of the MGD map of (a).

The connected components in the four clusters generated using MGD-based clustering are the potential text regions. As can be seen in Figure3.13(b), the text regions, especially the small text regions, include pixels from more than one cluster. If each cluster image is processed individually, some parts of text lines may be missed. Therefore, a multiple layer image generation strategy is required to keep the completeness of the text regions. Each layer image is composed of connected components that may be formed by text or non-text. All connected components should be processed until every CC is a single text line candidate are fed to a trained SVM classifier for text/non-text classification. Since the cluster with the smallest mean usually belongs to the background, CCMAP_1 is not considered in subsequent processes. The remaining three CC cluster maps are used to generate six layer images, denoted as $\mathit{LayerImg}_i(i = 1, \cdots, 6)$ in Equation 3.1.2. The layer images are demonstrated in Figure 3.5. The next step is to detect text by making use of these six layer images.

$$\begin{aligned} \mathit{LayerImg}_i &= \mathit{CCMAP}_{i+1}(i = 1, 2, 3), \\ \mathit{LayerImg}_4 &= \mathit{CCMAP}_2 + \mathit{CCMAP}_3, \\ \mathit{LayerImg}_5 &= \mathit{CCMAP}_2 + \mathit{CCMAP}_4, \\ \mathit{LayerImg}_6 &= \mathit{CCMAP}_3 + \mathit{CCMAP}_4. \end{aligned} \tag{3.1.2}$$



(a) Layer Image 1

(b) Layer Image 2

(c) Layer Image 3

(d) Layer Image 4

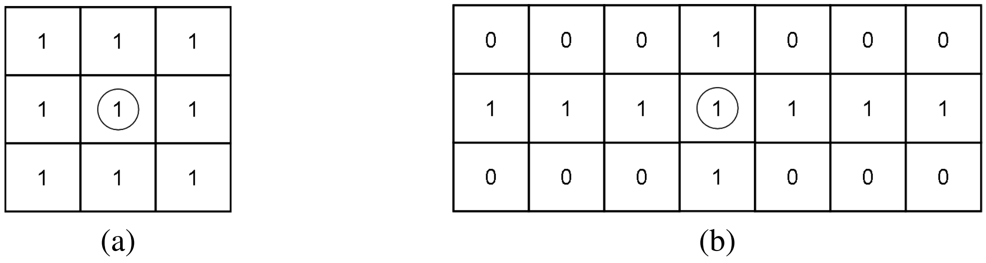
(e) Layer Image 5

(f) Layer Image 6

Figure 3.5: The layer images $LayerImg_i (i = 1, \cdots, 6)$ generated from Figure 3.13(c).

3.1.3 Morphological Operations

The six layer images are binary images containing possible text regions in the form of connected components. There are some non-text connected components to be filtered out. Meanwhile, the text connected components need to be recovered. Binary morphological operations are applied in our work to complete this job. Morphology is a technique for analysing and processing geometrical structures. In this section, only the basic operations: erosion, dilation and opening are discussed. Basically, morphology is to probe an image using a pre-defined shape on how this shape fits or misses the shapes in the image. This pre-defined shape is named as structuring element which is also a binary image. The results of morphological operations rely on the definition of the structuring element. Two factors determine a structuring element: shape and size. The shape could be a line, a square or a cross and so on. The size could be 3×3 or 2×5 and any predefined sizes.



1	1	1
1	①	1
1	1	1

(a)

0	0	0	1	0	0	0
1	1	1	①	1	1	1
0	0	0	1	0	0	0

(b)

Figure 3.6: (a) a 3×3 square-shape structuring element. (b) a 3×7 cross-shape structuring element.

Let E be an Euclidean space or an integer grid, A be a binary image in E and B be the structuring element.

- **Erosion**

  Erosion process makes thick connected components thinner. The erosion of a binary image A by the structuring element B is defined by:

  $$A \ominus B = \{z \in E | B_z \subseteq A\},$$

  where B_z is the translation of B by the vector z, that is $B_z = \{b + z | b \in B\}, \forall z \in E$.

- **Dilation**

  Dilation process makes thin connected components thicker. The dilation of A by the structuring element B is defined by:

  $$A \oplus B = \{z \in E | (B^s)_z \cap \ A \neq \emptyset\},$$

  where B^s is the symmetric of B, that is, $B^s = \{x \in E | -x \in B\}$.

- **Opening**

  Opening process can remove small and thin connected components. The opening of A by B is obtained by the erosion of A by B, followed by dilation of the resulting image by B:

$$A \circ B = (A \ominus B) \oplus B.$$

The implementations of morphological operations for precessing potential text connected components can be found in [33]. The dilation is performed first to link the character edges of every text line. Then, the opening was applied to suppress tiny components. In order to remove tiny connected components and line structures in the six layer images, morphological opening operation is performed with a cross-shape structuring element. The omission of the dilation is that the possible text components have been obtained after MGD-based clustering. The cross-shape structuring element applied is of the size of 3×7. The purpose is to remove the noise components with height less than 3 or width less than 7 is eliminated.

3.1.4 Cluster Post-processing

The vertical closeness of text lines may cause the connected components of text lines connect together vertically. Horizontal profile projection with connected component pixels is used to split complex connected components into individual text line connected components. Vertical profile projection using edge information is also implemented to separate the horizontally connected background from text. Then, the remaining connected components are enclosed by bounding boxes. To retrieve the missing parts of a text line on the top or at the bottom, we refine the bounding boxes by vertical expansion to make the text lines be included completely. A bounding box is expanded upward (or downward) if there are edge points above (or below) the top (or bottom) and within the horizontal range. The maximum expanded range is $1/3$ of the height of the bounding box on the top and at the bottom respectively. To this stage, each region enclosed by a vertically recovered bounding box will be sent to a trained classifier for classification which will be discussed in next section.

3.2 Fine Detection

The main purpose of a coarse detection is to obtain the bounding boxes enclosing text components rather than noise reduction. In spite of some non-text components are wiped off, there are still some false alarms. To solve this problem, a support vector machine classifier using a dedicated designed feature is resorted. This classifier is to identify whether the region inside each bounding box produced in the coarse detection step is a text box or not. In this section, the LBP-based features are discussed and a brief introduction of a support vector machine is presented.

3.2.1 T-LBP Descriptor

Local binary pattern (LBP) was first introduced by Ojala et al. [107] as a feature for texture classification. The invariance to monotonic gray-level changes and computational simplicity made LBP a popular texture classification method for a wide range of practical applications. The successful applications of LBP and its variants can be observed in many aspects of computer vision. These applications include face detection [108], face recognition [109], facial expression recognition [110], human detection [111], human action recognition [112] and gender classification [113]. A detailed survey on LBP based image classification can be found in [114].

The original LBP considers a 3×3 neighbourhood. The intensity of the central pixel is compared with those of its eight neighbour pixels. If a neighbour pixel has a greater or equal intensity value than that of the central pixel, this neighbour pixel is marked as '1'; otherwise, marked as '0'. Then, the LBP value of the central pixel is calculated as:

$$LBP(P_c) = \sum_{n=0}^{7} s(i_n - i_c)2^n, s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}, \tag{3.2.1}$$

where P_c denotes the central pixel and i_n and i_c denote the intensities of the n-th neighbour pixel and the central pixel. After all of the pixels in the image $f(x, y)$ are labelled with the corresponding LBP values, the histogram formulated by the LBP can be defined as

$$H_i = \sum_{x,y} I\{f(x, y) = i\}, i = 0, \cdots, n-1, \tag{3.2.2}$$

where n is the number of different LBP values, and I{A} is 1 if A is true and 0 if A is false.

In order to keep a uniform standard of LBP histograms generated from images with different sizes, LBP histograms must be normalised using Equation 3.2.3. A normalised LBP histogram represents the occurrence frequency of different LBP values.

$$N_i = \frac{H_i}{\sum_{j=0}^{n-1} H_j}. \tag{3.2.3}$$

Anthimopoulos et al. made the first attempt to modify the original LBP and proposed a new variant, eLBP, for text/non-text verification [115]. Later, Reduced eLBP and Multi-level Reduce eLBP were reported in their work as two advanced versions of eLBP [33]. In the particular case of capturing the characteristics of textual texture, eLBP deals with two important issues. The first one is that the traditional LBP operator generates quite different histograms for light texts on dark background and dark texts on light background. The second one is that LBP cannot describe the pattern of equal neighbour pixels with a same intensity. eLBP operator uses a parameter e to depict how different are the intensities between the centre pixel and a neighbour pixel are. In this way, eLBP captures the existence of an edge between the centre pixel and a neighbour pixel when the intensity difference is large enough. This is different from the original LBP operator which just simply compares which intensity is greater or smaller.

The formal definition of eLBP is as below:

$$eLBP(P_c) = \sum_{n=0}^{7} s_e(i_n - i_c)2^n, s_e(x) = \begin{cases} 1, |x| \geq e \\ 0, |x| < e \end{cases} . \tag{3.2.4}$$

In this thesis, a variant of LBP, known as T-LBP, is proposed to better depict the textual characteristic of text line. Experiments and observation have shown that text lines usually consist of horizontal and vertical strokes. This has motivated the use of the abundant vertical edges of text lines which usually show gradual changes of intensities along the horizontal direction at the edge of vertical strokes. The T-LBP describes these characteristics of texture of text lines as:

$$\begin{aligned} T\text{–}LBP_h(P_c) = \sum_{n=1}^{2} s_1(i_n - i_c)2^{n-1} + \sum_{n=3}^{4} s_2(i_n - i_c)2^{n-1}, \\ T\text{–}LBP_v(P_c) = \sum_{n=5}^{10} s_2(i_n - i_c)2^{n-5}, \\ s_1(x) = \begin{cases} 1, |x| \geq e_1 \\ 0, |x| < e_1 \end{cases} \quad and \quad s_2(x) = \begin{cases} 1, |x| \geq e_2 \\ 0, |x| < e_2 \end{cases} \end{aligned} \tag{3.2.5}$$

following the above notations in (3.2.1), where e_1=10, and e_2=20. T-LBP$_h$ is the T-LBP value in the horizontal direction and T-LBP$_v$ is the T-LBP value in the vertical direction. The neighbour assignment for T-LBP computation is shown in Figure 3.7.

The occurrence frequencies of T-LBP$_h$ values and T-LBP$_v$ values form two histograms. The dimension number of T-LBP$_h$ is $2^4 = 16$ and that of T-LBP$_v$ is $2^6 = 64$. Therefore, there are one 16-dimension feature vector and one 64-dimension feature vector. Catenated by these two feature vectors, the final feature vector is used to train an SVM classifier.

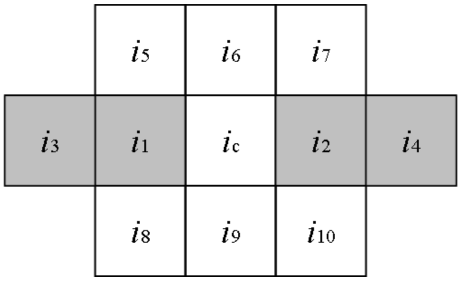


Figure 3.7: Neighbour assignment for T-LBP computation. The shadowed pixels represent horizontal neighbourhood pixels of the central pixel P_c.

3.2.2 IT-LBP Descriptor

The basic components of text are the strokes of characters. The orientations of the strokes include horizontal, vertical, diagonal and anti-diagonal directions. Different from the definition of T-LBP, IT-LBP takes diagonal and anti-diagonal directions into account as supplement. The local neighbourhood pixels of the four directions are illustrated in Figure 3.8 respectively. The pixels with shade in the 3×3 neighbourhood are the locations considered in computing the feature values.

The IT-LBP values of the central pixel at horizontal, vertical, diagonal and anti-diagonal directions are computed following Equations (3.2.6), (3.2.7), (3.2.8) and (3.2.9) respectively:

$$\text{IT-LBP}_h(P_c) = s_e(i_2 - i_1)2^0 + s_e(i_2 - i_3)2^1 + s_e(i_7 - i_6)2^2 + s(e)(i_7 - i_8)2^3 \quad (3.2.6)$$

$$\text{IT-LBP}_v(P_c) = s_e(i_4 - i_1)2^0 + s_e(i_4 - i_6)2^1 + s_e(i_5 - i_3)2^2 + s(e)(i_5 - i_8)2^3 \quad (3.2.7)$$

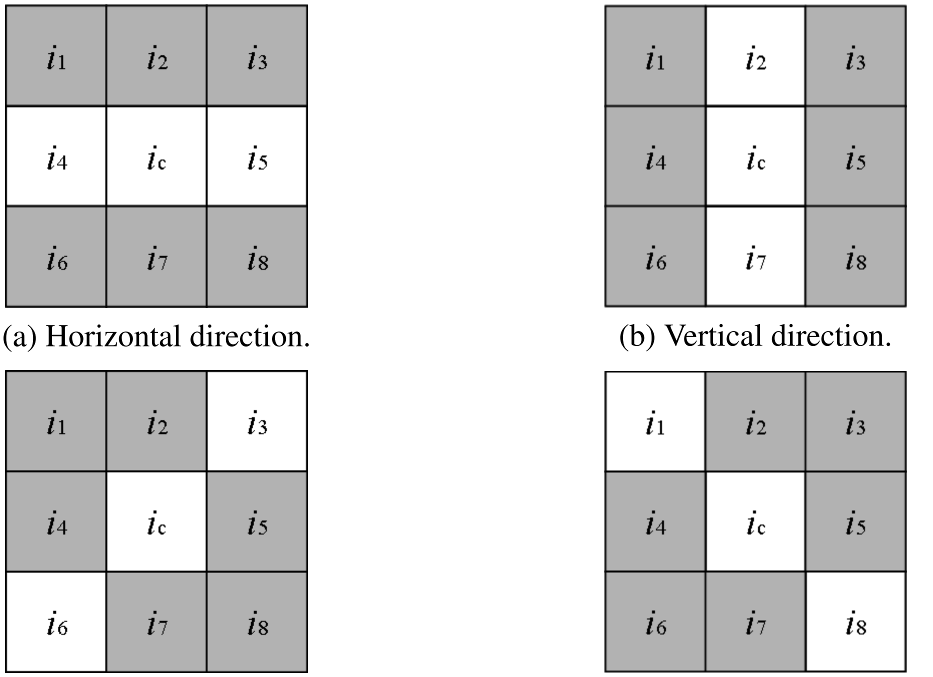
$$\text{IT-LBP}_d(P_c) = s_e(i_2 - i_5)2^0 + s_e(i_4 - i_7)2^1 + s_e(i_1 - i_8)2^2 \quad (3.2.8)$$

$$\text{IT-LBP}_{ad}(P_c) = s_e(i_2 - i_4)2^0 + s_e(i_5 - i_7)2^1 + s_e(i_3 - i_6)2^2 \tag{3.2.9}$$

where function s_e is defined as

$$s_e(x) = \begin{cases} 1, |x| \geq e \\ 0, |x| < e \end{cases}, \tag{3.2.10}$$

following the above notations in Equation (3.2.1).



(a) Horizontal direction.

(b) Vertical direction.

(c) Diagonal direction.

(d) Anti-diagonal direction.

Figure 3.8: The local neighbourhood of IT-LBP at four directions.

Each pixel has four IT-LBP values by considering horizontal, vertical, diagonal and anti-diagonal directions. The occurrence frequencies of IT-LBP$_h$ values, IT-LBP$_v$ values, IT-LBP$_d$ values and IT-LBP$_{ad}$ values construct four histograms. According to Equation (3.2.6) and Equation (3.2.7), IT-LBP$_h$ and IT-LBP$_v$ have $2^4 = 16$ dimensions respectively. According to Equation (3.2.8) and (3.2.9), IT-LBP$_d$ and IT-LBP$_{ad}$ have $2^3 = 8$ dimensions respectively. All of these four histograms are concatenated to form a $48-$dimensional feature vector. In our experiment, four different selections of e in Equation (3.2.10) are set to be 10, 30, 50 and 70. Therefore, the number of dimension of the IT-LBP feature used is of $48 \times 4 = 192$.

3.2.3 SVM-based Text/non-text Classification

The task of fine text region detection is to decide whether the region enclosed by each bounding box contains a text line or not. In this thesis, support vector machine (SVM) is used to solve a text/non-text classification problem. Thus, only the discussion on two-class SVM classification is included in this section. The original Support Vector Machine was introduced by Cortes and Vapnik [116]. Generally speaking, a support vector machine maps inseparable data into a higher-dimensional space to make the data separable. The function that splits the data into different classes in the higher-dimensional space is called hyperplane. Since a support vector machine is a supervised learning model, a group of training data is used to train the support vector machine. The parameters embedded in the support vector machine are determined based on the given training data. Another group of data are used to test the trained support vector machine. Support vector machine is a commonly used tool in pattern recognition. The applications of support vector machine for two-class classification can be seen in various text detection methods [33, 115].

SVM of linearly separable data

In a two-class linearly separable classification problem, all of the feature vectors $\boldsymbol{x}_i, i = 1, 2, \cdots, N$, which are also called training samples, in the training set X are classified into two classes, $C1$ and $C2$ by a hyperplane [117]:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 = 0. \tag{3.2.11}$$

where $\boldsymbol{w}$ and w_0 are the direction and the position of hyperplane respectively.

There are numerous possible choices of the hyperplane to bi-separate all of the training samples. As illustrated in Figure 3.9, each point stands for a training sample and each line stands for a feasible hyperplane. Here, all of the three straight lines can clearly separate them into two groups. Superficially, any possible hyperplane can be a solution. However, there is a big concern that whether any hyperplane can correctly classify unknown data outside the training set. This is a very important issue which is known as the generalisation performance of a classifier. A classifier with a perfect performance in classifying training samples may have a poor capability in classifying unknown data. This phenomenon is called overfitting. A classifier with good generalisation ability can prevent overfitting. Support vector machine seeks the optimal hyperplane that can give the maximum margin from $C1$ and $C2$ equally.

A hyperplane is determined by its direction $\boldsymbol{w}$ and its position w_0. The optimal hyperplane should have the same distance from the closest points in $C1$ and $C2$ respectively. The distance between a hyperplane and a point is defined by

$$d = \frac{|g(\boldsymbol{x})|}{||\boldsymbol{w}||}. \tag{3.2.12}$$

By scaling $\boldsymbol{w}$ and w_0, $g(\boldsymbol{x})$ is equal to 1 and -1 at the closest points in $C1$ and $C2$ respectively. This condition is equivalent to

$$\begin{gathered}\text{The margin is } \frac{1}{||\boldsymbol{w}||} + \frac{1}{||\boldsymbol{w}||} = \frac{2}{||\boldsymbol{w}||} \text{ subject to} \\ \boldsymbol{w}^T\boldsymbol{x} + w_0 \geq 1, \forall \boldsymbol{x} \in C1 \text{ and } \boldsymbol{w}^T\boldsymbol{x} + w_0 \leq -1, \forall \boldsymbol{x} \in C2.\end{gathered} \tag{3.2.13}$$

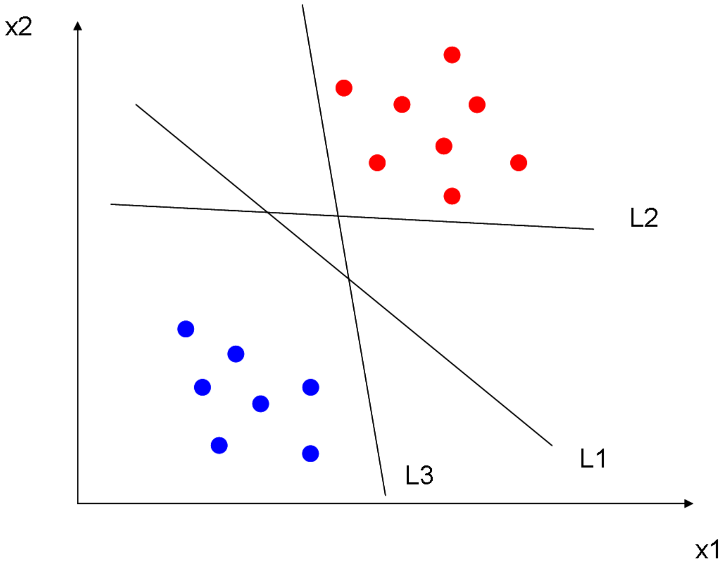


Figure 3.9: Possible hyperplanes in a linearly separable case. The red and blue points represent training samples belonging to C1 and C2 respectively. The straight lines L1, L2 and L3 are capable to separate the points into two groups.

For each $\boldsymbol{x}_i$, the corresponding class is denoted by y_i (+1 for $C1$ and -1 for $C2$). Then, Equation 3.2.13 is converted to the following minimisation problem:

$$\text{minimise } J(\boldsymbol{w}, w_0) = \frac{1}{2}||\boldsymbol{w}||^2 \tag{3.2.14}$$

$$\text{subject to } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1, i = 1, 2, \cdots, N. \tag{3.2.15}$$

All of the feature vectors that satisfy the equalities of Equation 3.2.15 are called support vectors. Support vectors are crucial elements in the training set that decide the direction and position of the optimal hyperplane.

In order to minimise Equation 3.2.14, the following Karush-Kuhn-Tucker (KKT) conditions [117] should be satisfied:

$$\frac{\partial L(\boldsymbol{w}, w_0, \lambda)}{\partial \boldsymbol{w}} = \boldsymbol{0} \tag{3.2.16}$$

$$\frac{\partial L(\boldsymbol{w}, w_0, \lambda)}{\partial w_0} = 0 \tag{3.2.17}$$

$$\lambda_i \geq 0, i = 1, 2, \cdots, N \tag{3.2.18}$$

$$\lambda_i [y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) - 1] = 0, i = 1, 2, \cdots, N \tag{3.2.19}$$

where λ is the vector of the Lagrange multipliers, λ_i, and $L(\boldsymbol{w}, w_0, \lambda)$ is the Lagrangian function defined as

$$L(\boldsymbol{w}, w_0, \lambda) = \frac{1}{2}\boldsymbol{w}^T \boldsymbol{w} - \sum_{i=1}^{N} \lambda_i [y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) - 1]. \tag{3.2.20}$$

Combining Equation 3.2.20 with Equation 3.2.16 and Equation 3.2.17 results in

$$\boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i \tag{3.2.21}$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0. \tag{3.2.22}$$

Substituting Equation 3.2.21 and Equation 3.2.22 into Equation 3.2.20

$$L(\boldsymbol{w}, w_0, \lambda) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j. \tag{3.2.23}$$

Considering Equation 3.2.21 and Equation 3.2.22, the optimal Lagrange multipliers λ_i^* ($i = 1, 2, \cdots, N$) can be obtained by maximising Equation 3.2.23. This problem can be solved by quadratic programming (QP). After all Lagrange multipliers are fixed, the optimal $\boldsymbol{w}^*$ and w_0^* can be figured out. Finally, the optimal hyperplane is obtained and it is unique (Figure 3.10). For every unknown datum $\boldsymbol{x}$, the class it belongs to is decided by the following equation:

$$f(\boldsymbol{x}) = sgn(\boldsymbol{w}^{*T} \boldsymbol{x} + w_0^*) \tag{3.2.24}$$

where

$$sgn(x) = \begin{cases} 1, \text{ if } x > 0 \\ -1, \text{ else.} \end{cases} \tag{3.2.25}$$

Hence, when an unknown datum is fed to the trained SVM classifier, it is classified as positive (or negative) class if the result of Equation 3.2.24 is +1 (or -1).

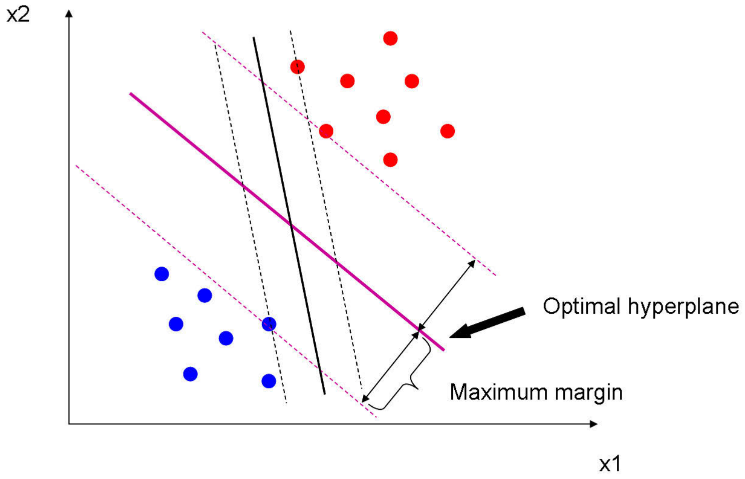


Figure 3.10: Optimal SVM hyperplane having the maximum margin.

SVM of linearly inseparable data

The above discussion on support vector machine is based on the assumption that the training data are linearly separable. However, the training data may be linearly inseparable in practice. This means that no matter how the hyperplane is drawn, there are always some positive training data partitioned into the negative class or vice versa. An example is depicted in Figure 3.11. To solve this problem, support vector machine can be extended by adding slack variables $\xi_i(\geq 0)$ and a penalty factor C into Equation 3.2.14. For obtaining the optimal hyperplane, the target is below:

$$\text{minimise } J(\boldsymbol{w}, w_0, \xi) = \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{N}\xi_i \tag{3.2.26}$$

$$\text{subject to } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 - \xi_i, i = 1, 2, \cdots, N \tag{3.2.27}$$

$$\xi_i \geq 0, i = 1, 2, \cdots, N. \tag{3.2.28}$$

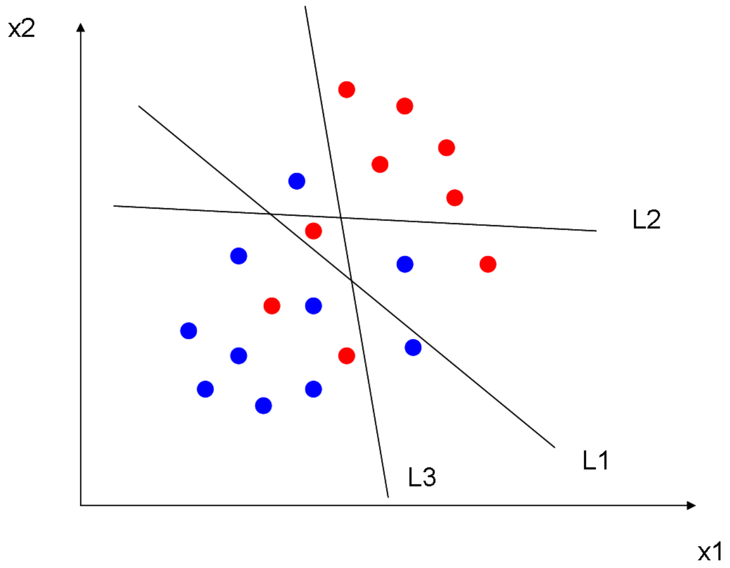


Figure 3.11: Linearly inseparable training data. Any straight lines cannot separate all of the red and blue points into the group they belong to.

Accordingly, the corresponding Lagrangian function becomes

$$\begin{aligned} L(\boldsymbol{w}, w_0, \lambda) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i \\ - \sum_{i=1}^{N}\lambda_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) - 1 + \xi_i] \end{aligned} \tag{3.2.29}$$

The corresponding Karush-Kuhn-Tucker conditions are

$$\frac{\partial L}{\partial \boldsymbol{w}} = \mathbf{0} \text{ or } \boldsymbol{w} = \sum_{i=1}^{N}\lambda_i y_i \boldsymbol{x}_i \tag{3.2.30}$$

$$\frac{\partial L}{\partial w_0} = 0 \text{ or } \sum_{i=1}^{N}\lambda_i y_i = 0 \tag{3.2.31}$$

$$\frac{\partial L}{\partial \xi_i} = 0, i = 1, 2, \cdots, N \tag{3.2.32}$$

$$\lambda_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) - 1 + \xi_i] = 0, i = 1, 2, \cdots, N \tag{3.2.33}$$

$$\mu_i\xi_i = 0, i = 1, 2, \cdots, N \tag{3.2.34}$$

$$\mu_i \geq 0, \lambda_i \geq 0, i = 1, 2, \cdots, N. \tag{3.2.35}$$

Substituting Equation 3.2.30 and Equation 3.2.31 into Equation 3.2.29, the corresponding quadratic programming (QP) problem becomes

$$\text{maximise} \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{3.2.36}$$

$$\text{subject to } 0 \leq \lambda_i \leq C, i = 1, 2, \cdots, N \tag{3.2.37}$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0. \tag{3.2.38}$$

In Equation 3.2.36, function $K(\cdot)$, called kernel function, is a dot-product function that maps the linearly inseparable training data into a high-dimensional feature space to increase linear separability. The commonly used kernel functions are

- Radial basis function (RBF): $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(= -\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||), \gamma > 0$,
- Linear: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$,
- Polynomial: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\gamma \boldsymbol{x}_i^T \boldsymbol{x}_j + r)^d$,
- Sigmoid: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = tanh(\gamma \boldsymbol{x}_i^T \boldsymbol{x}_j + r)$.

Here, γ, r and d are parameters.

3.2.4 Bounding Box Integration

In each layer image, the verified text lines are enclosed by bounding boxes. All of the bounding boxes from all layer images are integrated together to form the final bounding boxes. Since a single text line may appear in more than one layer image, the purpose of bounding box integration is to combine the overlapping bounding boxes. Let $B1$ and $B2$

stand for two bounding boxes. If the criterion in Equation 5.2.6 is met, $B1$ and $B2$ are integrated into one bounding box:

$$\frac{AREA(B1 \cap B2)}{min(AREA(B1), AREA(B2))} > 0.6, \tag{3.2.39}$$

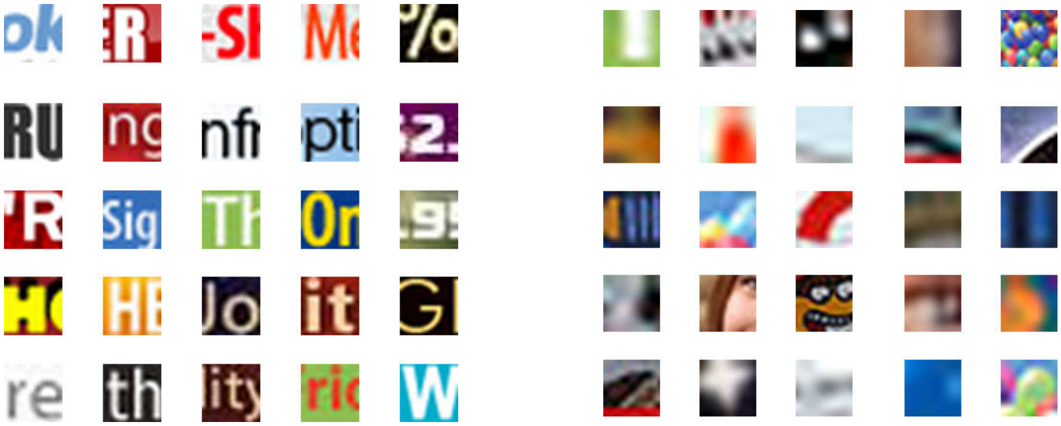
where $AREA(B1 \cap B2)$ stands for the overlapping areas of $B1$ and $B2$. $AREA(B1)$ and $AREA(B2)$ are the area of $B1$ and $B2$ respectively.

3.3 Experimental Results

In this section, the discriminating capability of different LBP-based features and various text detection methods are compared. The classification performance of the classifiers trained by the proposed features is compared with the classifiers trained by other LBP-based features. The superiority of the proposed text detection algorithm over other algorithms is illustrated by using a standard born-digital text detection dataset.

3.3.1 Comparison of Different LBP-based Features

In order to demonstrate the superiority of the proposed features, the performance of the SVM classifiers trained using different features are compared. Both training samples and testing samples are taken from the ICDAR2011 born-digital text localisation dataset. There are a total of 420 training images included in the training set and 102 test images in the test set. In order to obtain the training samples, the coarse detection is applied to the 420 training images leading to the candidate text lines. These candidate text lines are further divided into samples with the size of 20×20. In this way, 9000 positive samples and 11929 negative training samples are generated. Some examples of positive samples and negative samples are shown in Figure 3.12. All of these 20929 samples are separated into two groups: training group and testing group. There are 6000 samples randomly selected to be placed in the training group and the remaining 14929 samples are used as the testing group. Some examples of positive training samples and negative training samples are shown in Figure 3.12. LIBSVM [118] is employed for training the SVM classifiers with different features. The radial basis function kernel is adopted and the optimal parameters, σ and γ, are obtained by grid search. The comparison results on the test samples are listed in Figure 3.1. The 102 testing images are used for testing the performance of our algorithm which is compared with other existing text detectors in Section3.3.2.



(a)
(b)

Figure 3.12: Examples of training samples. (a) Positive samples. (b) Negative samples.

Accuracy Rate (AR), Text Recall (TR), Text Precision (TP) and Non-text Error Rate (NTER) are used to evaluate the power of different classifiers on discriminating text and non-text samples. The definitions of the above four criteria are given as follows:

$$\mathrm{AR} = \frac{N_{ctn}}{N_{tn}} \times 100\% \tag{3.3.1}$$

$$\mathrm{TR} = \frac{N_{cct}}{N_{t}} \times\ 100\% \tag{3.3.2}$$

Feature	AR	TR	TP	NTER
IT-LBP	95.25%	93.10%	95.73%	3.13%
T-LBP	92.98%	92.36%	91.40%	6.56%
eLBP [115]	92.89%	89.80%	93.42%	4.77%
LBP [107]	90.26%	92.93%	85.63%	11.76%

Table 3.1: Comparison on classification performance of different LBP-based features.

$$\text{TP} = \frac{N_{cct}}{N_{ct}} \times 100\% \tag{3.3.3}$$

$$\text{NTER} = \frac{N_{wcnt}}{N_{nt}} \times 100\% \tag{3.3.4}$$

where N_{ctn} is the number of correctly classified text and non-text samples, N_{tn} means the number of text and non-text samples, N_{cct} stands for the number of correctly classified text samples, N_t is the number of text samples, N_{ct} represents the number of samples classified as text, N_{wcnt} is the meaning of the number of non-text samples wrongly classified as text and N_{nt} means the number of non-text samples.

3.3.2 Results Obtaining by Using Public Dataset

In order to determine whether a candidate text block contains text or not in the fine detection step, it is firstly normalised to 20 pixels in height keeping its aspect ratio unchanged. For each text block candidate, the feature values are calculated and an 80-dimension feature vector is formed. Then, the feature vector is fed into a trained SVM classifier to verify whether the text block candidate contains text or not. The verification of each normalised bounding box proceeds using a 20×20 sliding window with a 4-pixel step. The classifier is trained by using the scheme described in Section 3.3.1 and is then used to classify whether

Method	Recall	Precision	Harmonic Mean
Ours with IT-LBP	**75.96%**	**87.75%**	**81.43%**
Ours with T-LBP	**74.88%**	**85.35%**	**79.78%**
Textorter [1]	69.62%	85.83%	76.88%
TH-TextLoc [1]	73.08%	80.51%	76.62%
TDM_IACAS [1]	69.16%	84.64%	76.12%
OTCYMIST [1]	75.91%	64.05%	69.48%
SASA [1]	65.62%	67.82%	66.70%
Text Hunter [1]	57.76%	75.52%	65.46%

Table 3.2: Comparisons between our method and the algorithms in ICDAR2011 Robust Reading Competition Challenge 1 [1].

each scanning window is text or not. The SVM decision value $G(z)$ of each sliding window is accumulated when the sliding window moves along a candidate text block. The confidence $Conf(R)$ of a candidate text line R is computed by using the definition in [32]:

$$Conf(R) = \sum_{z \subseteq R} G(z) \cdot \frac{1}{\sqrt{2\pi}\sigma_0} exp\left(\frac{d_z^2}{2\sigma_0^2}\right), \tag{3.3.5}$$

where d_z is the distance between the center of window z and the center of the text region R, and $\sigma_0 = 10$. A candidate text line R is identified as a text line if $Conf(R) \geq 0$.

The proposed algorithm is tested against the ICDAR2011 born-digital image dataset, which is made and published for the ICDAR 2011 Robust Reading Competition Challenge 1: Reading Text in Born-Digital Images [1]. In order to compare with other algorithms under the same condition, all of the 102 test images and the same performance evaluation system [119] are used in the competition. Results in the competition (see [1] for more details) are illustrated in Table 3.2.

Some detection results are shown in Figure 3.13. High contrast of text is caused by opposite shades of text and background, such as the pink texts on white background in Figure 3.13(a) and the white texts on black background in Figure 3.13(f). Low contrast of text is due to similar shades of text and background, for example, the gray texts on white background in Figure 3.13(a) and the purple texts on black background in Figure 3.13(e). The examples of detection results in Figure 3.13 illustrate that the proposed method can detect text lines with high contrast and low contrast. Also, it can be seen that the classifier trained by the T-LBP can greatly remove the non-text areas caused by complex background, which can be observed in Figures 3.13(f) and (g).

3.4 Summary

In this chapter, a text detection algorithm containing coarse detection and fine detection is developed. Using binary edge information and the magnitude of gradient for describing text regions is compared. Binary edge extraction is quite sensitive to threshold setting. The advantage of using magnitude of gradient over binary edge is that the gradient information is not removed by thresholding. The existing MGD-based text detection approaches are analysed and a new strategy which creates multiple layer images from an MGD map is further developed. With this strategy, text lines with both low and high contrasts can be detected in the coarse detection step. After that, some morphological operations and post-processing are employed to generate text line candidates for fine text detection. In order to eliminate false alarms, a supervised machine learning using a newly developed feature is used. LBP, a texture-based feature, and its variant eLBP are discussed. Combining discriminative power of LBP-based textural feature and the stroke structure of text lines, two variants of LBP, T-LBP and IT-LBP, are proposed. The superior performance of the SVM classifiers trained with T-LBP and IT-LBP are demonstrated by comparing with the classifiers obtained with LBP and eLBP. The overall detection results using the proposed framework are shown in the comparison with other algorithms.

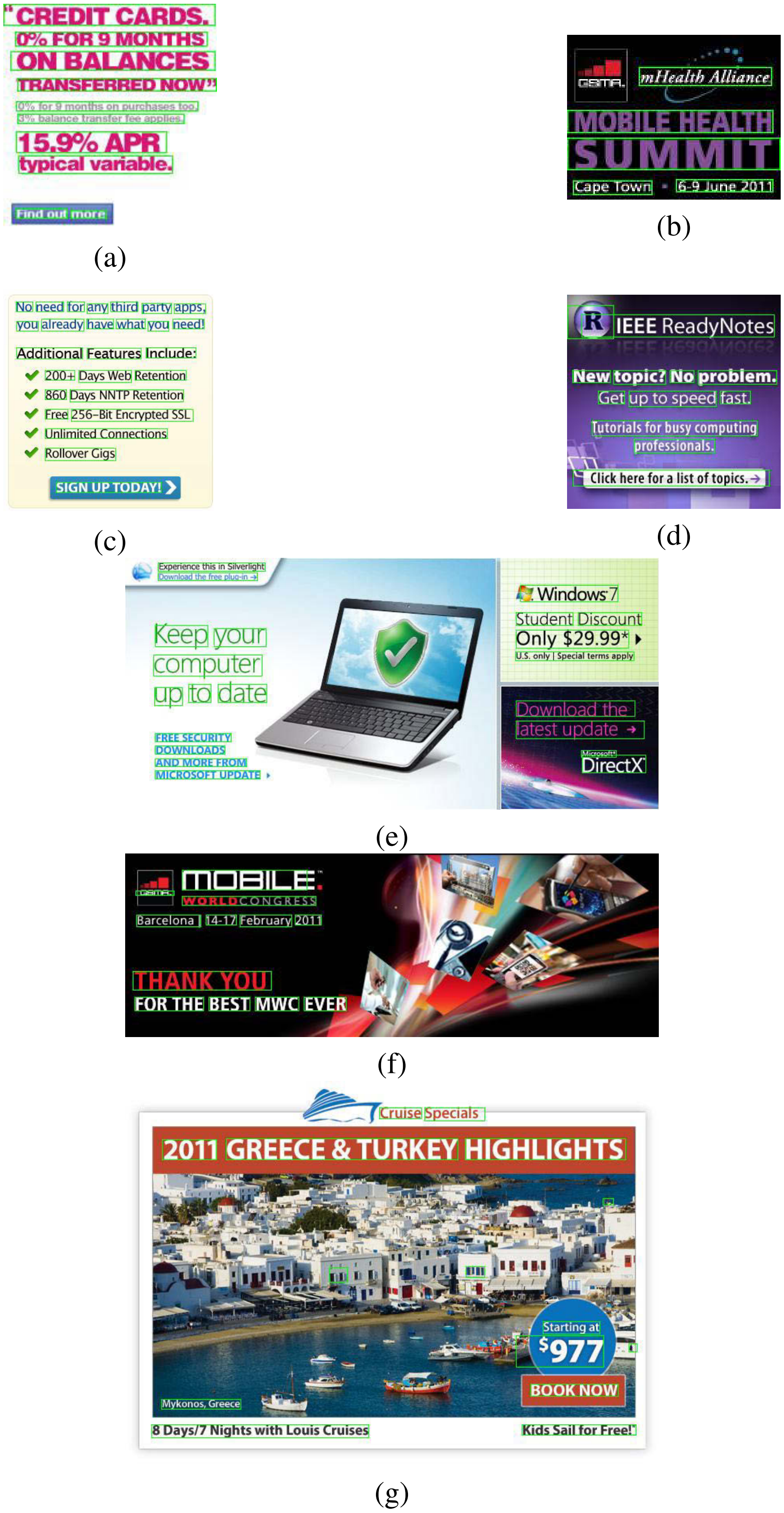


Figure 3.13: Some text detection results by the proposed method (the detection results are shown in green bounding boxes).

Chapter 4

Natural Scene Text Detection

In the previous chapter we have presented an effective algorithm for detecting text lines from born-digital images. The multiple layer image strategy enables the algorithm to detect text lines with both strong and weak contrasts, which stem from the transition between text and background at the boundary of text strokes. The generation of the multiple layer images is based on the Maximum Gradient Difference (MGD) computed within the local neighbourhood of every pixel. The possible text lines with different contrasts appear in different layer images in the form of connected components. The success of text detection relies on that the local neighbourhood used for computing the MGD values can cover the range of multiple characters. Since the text lines in born-digital images tend to have narrow gaps between characters, the region of a text line can be easily connected together by MGD-based clustering and morphological operations. However, for text in natural scene images, characters typically have large sizes, wide stroke width and big inter-character gaps. All of these can lead to detection failures when adopting the framework presented in the previous chapter for detecting natural scene text. In this chapter, each character is treated as an independent object and all character objects are grouped into text lines.

The framework of our natural scene text detection algorithm is illustrated in Figure

4.1. Maximally Stable Extremal Regions (MSER) [120] are extracted on the gray level image to generate character candidates. Both dark-on-bright MSERs and bright-on-dark MSERs are obtained due to the existence of texts with two polarities. In order to remove the character MSERs and keep the non-character MSERs simultaneously, a supervised machine learning stage uses a set of features to train a classifier for character/non-character MSER classification. In order to bring back the misclassified character MSERs, an MSER retrieval step is applied to retrieve single character MSERs and multiple character MSERs. Then, the remaining MESRs are grouped into text lines. As there are still a large amount of non-text regions, a text/non-text line classification step is performed to eliminate the false alarms. A bootstrap scheme is also utilised to enhance the capability of eliminating non-text regions. Finally, the same evaluation framework and dataset in ICDAR2011 Text Localisation Competition [2] are used to evaluate the proposed algorithm.

The remainder of this chapter is organised as follows. Section 4.1 presents the generation of character MSERs from the original natural scene images as character candidates. The features used for training a classifier to classify character MSERs and non-character MSERs are discussed in Section 4.2. Retrieving both single character MSERs and multiple character MSERs are given in Section 4.3. Section 4.4 talks about how the classified character MSERs are grouped into text lines. This is followed by the presentation of false alarm elimination in Section 4.5. The experimental results are shown in Section 4.6. Section 5.4 summarises this chapter.

4.1 Character MSER Generation

This initial step of this algorithm is to obtain text candidates from natural scene images. The recent research [25, 26, 60, 121] on natural scene text detection shows that connected

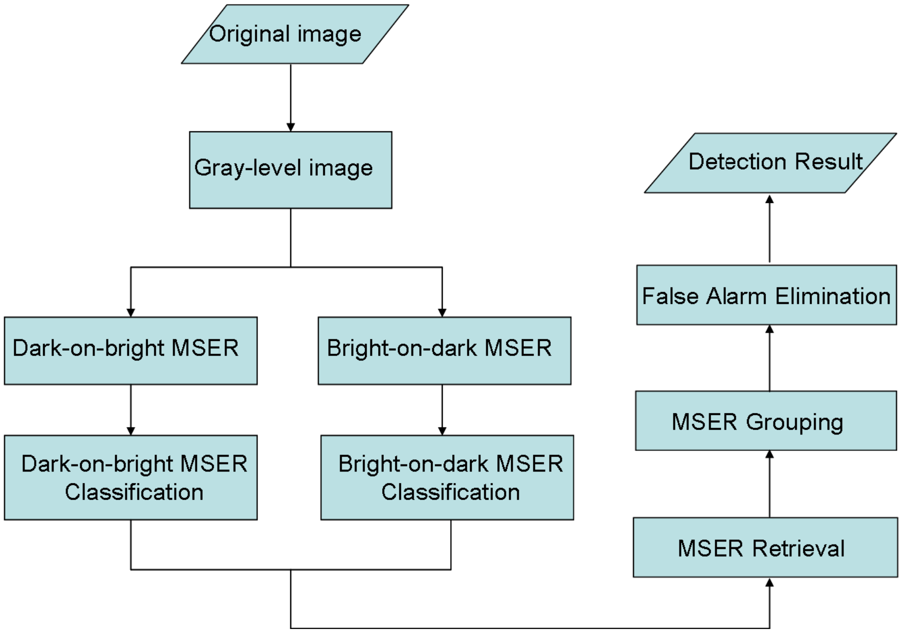


Figure 4.1: The framework of the proposed natural scene text detection algorithm.

component (CC) analysis based on character strokes as an initial processing stage is an attractive solution. In [25], Epshtein et al. performed stroke width transform on the Canny edge maps of scene images. Pixels with similar stroke width were combined together to form CCs. Yi et al. [26] proposed gradient-based partition and colour-based partition to generate CCs from scene images. In the work of Pan et al. [60, 121], a local binarisation algorithm was applied to obtain CCs as candidate text components for further processing. The principle of the methods belonging to this category considers that every character is an individual CC and the character CCs are grouped into text lines after the non-character CCs are eliminated. An advantage of this type of method is that the character CCs are less likely to connect to the background components. By taking this advantage, the proposed algorithm belongs to this category of text detection algorithms.

After the comparison with other region detectors as published in [122], Maximally Stable Extremal Regions (MSERs) detector has been acknowledged as one of the best region detectors as it is robust to view point, scale and lighting changes. An MSER is a part of the image where local binarisation is stable over a large range of thresholds. For integrity, the definition of MSER is introduced by following the notations in [120].

Image I is a mapping $I : D \subset Z^2 \rightarrow S$. Extremal regions are well defined on images if

1. S is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation $\leq$ exists. Only $S = \{0, 1, \cdots, 255\}$ is considered in our algorithm.

2. An adjacency relation $A \subset D \times D$ is defined. $p, q \in D$ are adjacent (i.e. pAq) if and only if $\sum\limits_{i=1}^{d} |p_i - q_i| \leq 1$. d is the dimension of p and q.

Region Q is a contiguous subset of D, i.e. for each $p, q \in Q$ there is a sequence $p, a_1, a_2, \cdots, a_n, q$ and $pAa_1, \cdots, a_iAa_{i+1}, \cdots, a_nAq$ where $a_j(j = 1, \cdots, n) \in Q$. Region boundary $\partial Q = \{q \in D \setminus Q : \exists p \in Q : qAp\}$, i.e. the boundary ∂Q of Q is the set of pixels being adjacent to at least one pixels of Q but not belonging to Q. Extremal region $Q \subset D$ is a region such that for all $p \in Q$ and $q \in \partial Q, I(p) > I(q)$ (maximum intensity region) or $I(p) < I(q)$ (minimum intensity region). Let $Q_1, \cdots, Q_{i-1}, Q_i, \cdots$ be a sequence of nested extremal regions, i.e. $Q_i \subset Q_{i+1}$. Q_{i^*} is maximally stable extremal region (MSER) if and only if $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}|/|Q_i|$ has a local minimum at i^* ($| \cdot |$ denotes cardinality). $\Delta \in S$ is a parameter.

Natural scene characters typically have strong contrast against background, uniform colour and hence uniform intensity. Therefore, pixels belonging to a character can be united as an MSER when they are extracted from the gray-level map. In the implementation of MSER generation, the resultant regions can be either of two types: bright regions on dark background and dark regions on bright background. Dark-on-bright MSERs and bright-on-dark MSERs are processed separately since both of them appear in natural scene images.

Some examples of MSER extraction from natural scene images are shown in Figure 4.2 and some extracted character MSERs and non-character MSERs are illustrated in Figure 4.3 and Figure 4.4 respectively.

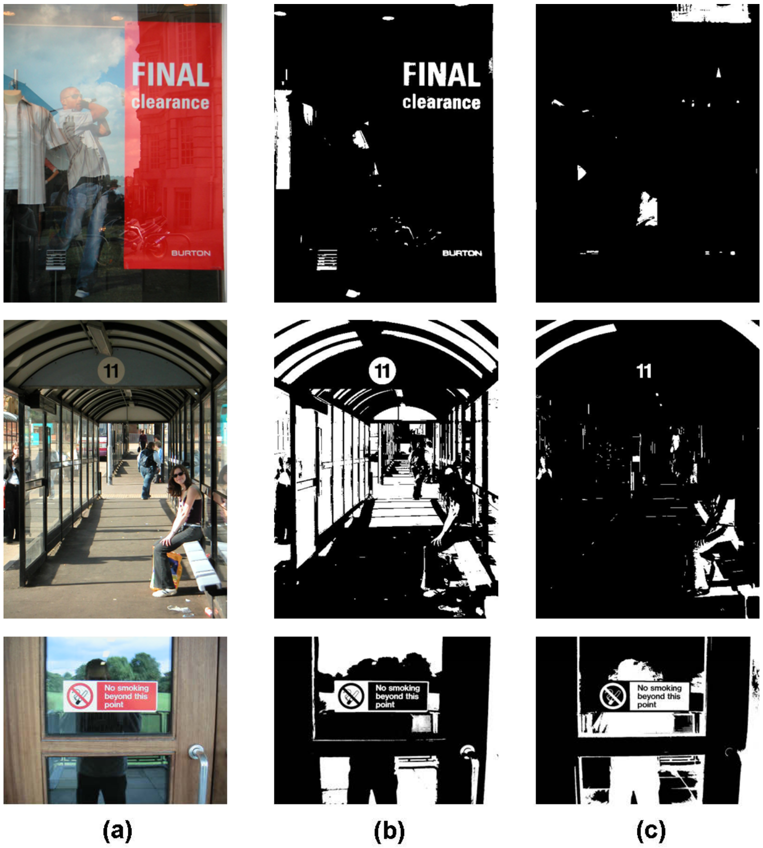


Figure 4.2: MSER extraction results. MSER regions are marked in white and the remaining regions are marked in black. (a) Original images. (b) Bright-on-dark MSERs. (c) Dark-on-bright MSERs.

Before proceeding to the next stage, the MSERs that are obviously not character MSERs are filtered out. MSERs that are too small or too high are removed as non-characters. MSERs with the height less than 10 pixels are also discarded. As only the upper and lower case Roman letters and arabic numbers are considered, the maximum number of holes of a character MSER is 2 as in "B", "g" and "8" for examples. If an MSER has more than

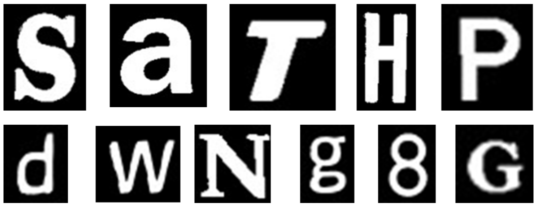


Figure 4.3: Character MSER samples.

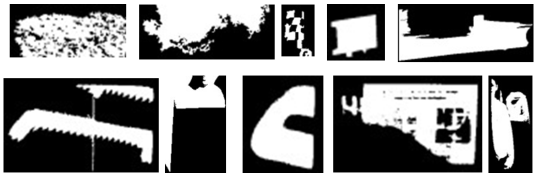


Figure 4.4: Non-character MSER samples.

2 holes, they are pruned as well. The remaining MSERs are fed into a MSER classifier trained by the features introduced in the next section for text/non-text MSER classification.

4.2 Character MSER Features

In the previous step, both bright-on-dark MSERs and dark-on-bright MSERs are generated. However, character MSERs as well as non-character MSERs are extracted. The non-character MSERs are not the objects-of-interest and give a very negative influence on the later steps. Therefore, suppressing the non-character is the first concern. In order to remove the non-character MSERs and keep the character MSERs simultaneously, a supervised machine learning scheme is used to train a classifier for discriminating character MSERs from non-character MSERs. Four types of features that can embody the characteristics of character MSERs are employed. These are geometry-based, stroke-based, gradient-based and colour-based features. Geometry-based features describe the geometric properties. Stroke-based features investigate the uniformity of stroke width in characters. Gradient-based features depict the double edges of opposite gradient directions that character strokes have. The colour-based feature used is the variance of local foreground/background colour difference at stroke edges.

4.2.1 Geometry-based Features

According to observation, character MSERs normally have more regular shapes than non-character MSERs. Therefore, aspect ratio, occupation ratio, regularity and compactness are employed to describe the geometric properties of character MSERs.

- **Aspect ratio**. This feature intends to remove non-character MSERs which are too long or too narrow because the aspect ratios of character MSERs are in a limited range. The definition of aspect ratio (AR) is

  $$\text{AR}(M) = min(\frac{w}{h}, \frac{h}{w}), \tag{4.2.1}$$

  where w and h are the width and height of the MSER M respectively.

- **Occupation ratio**. Pixels belonging to a character MSER normally do not occupy too much or too less area within its bounding box. Occupation ratio (OR) is defined as

  $$\text{OR}(M) = \frac{p}{w \times h}, \tag{4.2.2}$$

  where p represents the number of pixels belonging to MSER, and w and h are defined in Equation 4.2.1.

- **Regularity**. Usually, the skeleton points of a character MSER lie nearly along the center of character strokes. The ratio between the number of skeleton pixels and the number of stroke contour pixels of a character MSER tends to be stable for different characters with various fonts. The definition of regularity (R) is

$$\mathrm{R}(M) = \frac{N_{skel}}{N_{con}}, \tag{4.2.3}$$

where N_{skel} is the number of points on skeleton and N_{con} is the number of points on the stroke contour.

- **Compactness**. Non-character MSERs with too complex contour shape ought to be erased. Compactness (C) is defined as

$$\mathrm{C}(M) = \frac{Area}{N_{con}^2}, \tag{4.2.4}$$

where $Area$ is the area of the bounding box of an MSER and N_{con} is the number of contour pixels of the MSER.

4.2.2 Stroke-Based Features

A character is composed of strokes and this characteristic leads to the exploration of features based on strokes. Stroke-based features stem from the approximately uniform stroke width and double edges of opposite gradient directions. These two aspects were considered together in the work of Zhang et al. [52] and the concept of “Character Energy” was conceived. After obtaining binary edge points from the gray-level image, the character energy of the edges of each possible character was used in an unsupervised framework to classify characters and non-characters. “Character Energy” is applied to MSERs to assess the possibility of an MSER being a character MSER. Furthermore, since the contours of character MSERs are closed, the computation of MSER character energy is more reliable than that of character energy from binary edge points. The reason is that character energy would be erroneously computed when the edges of a character are broken.

Taking dark characters on bright background into consideration, since a character MSER has a complete contour, a ray generated from a contour point, named as a source point, travelling along the gradient direction of the point can reach another contour point. If the reached contour point has an approximately opposite gradient direction, it is called a sibling point of the source point. The distance between the source point and its sibling point is the stroke width. Since machine-printed characters typically have uniform stroke width, this distance tends to be similar at different contour points of a character. Here, we consider both opposite gradient direction and uniform stroke width as features to differentiate character MSERs from non-character MSERs.

- **Average Gradient Direction Difference of Sibling Point Pairs**. For a character MSER, most contour points can find their sibling points. Each contour point and its sibling point forms a sibling point pair. According to the definition of sibling points, the absolute value of the difference of the gradient directions of a sibling pair is close to π. By taking all contour points into account, the absolute average difference of gradient direction of all sibling pairs should be near π. Let N denote the number of contour points of an MSER, $P^{(i)}$ denote the i-th contour point of a character MSER with the sibling point $SP^{(i)}$, and $\theta_P^{(i)}$ and $\theta_{SP}^{(i)}$ denote the gradient directions at $P^{(i)}$ and $SP^{(i)}$. The difference of gradient direction of $P^{(i)}$ is defined as follows:

  $$\theta_{angle}^{(i)} = \text{abs}(\pi - \text{abs}(\theta_P^{(i)} - \theta_{SP}^{(i)})), \tag{4.2.5}$$

  where abs(x) is the absolute value of x. When the gradient direction of $P^{(i)}$ is opposite to the gradient direction of $SP^{(i)}$, $\theta_{angle}^{(i)} = 0$. The Average Gradient Direction Difference for a contour α is defined as:

  $$\alpha = \frac{\sum\limits_{i=1}^{N} \theta_{angle}^{(i)}}{N}. \tag{4.2.6}$$

For character MSERs, α should be a value close to 0.

- **Ratio of Sibling Point Pairs**. As most contour points have sibling points, the proportion of sibling point pairs in a character MSER should be high. The fraction of sibling point pairs η_{sp} is defined as below:

$$\eta_{sp} = \frac{\sum_{i=1}^{N} h(\theta_{angle}^{(i)}, \frac{\pi}{6})}{N}, \tag{4.2.7}$$

where

$$h(\theta_{angle}^{(i)}, \frac{\pi}{6}) = \begin{cases} 1, \theta_{angle}^{(i)} \leq \frac{\pi}{6} \\ 0, else. \end{cases} \tag{4.2.8}$$

It can be seen from Equation 4.2.7 that the more sibling pairs an MSER has, the greater η_{sp} is and, therefore, the more possible the MSER to be a character.

- **MSER Character Energy**. According to the definitions of α and η_{sp}, a character MSER tends to have a low α and a high η_{sp}. By combining α and η_{sp} together, MSER Character Energy can be defined to give the possibility that an MSER is a character MSER. The function should be proportional to η_{sp} and anti-proportional to α. We define MSER Character Energy E as

$$E = (\frac{\pi - \alpha}{\pi} + \eta_{sp})/2. \tag{4.2.9}$$

According to the definition of character MSER energy, $E \in [0, 1]$, and the closer α is to 0 and the closer η_{sp} is to 1, the closer E is to 1. An MSER with greater E has higher probability to be a character MSER.

- **Ratio of Dominant Stroke Width**. The distance between a source point $P^{(i)}$ and its sibling point $SP^{(i)}$ can be considered as a possible stroke width. Let $SW =$

$\{sw^{(j)}|j = 1, \cdots, J\}$ be the collection of all stroke width values $sw^{(j)}$ of an MSER and $hist(SW)$ be the histogram of SW, where J is the total number of different stroke width values. For a character MSER, typically there is a stroke width value that has the greatest occurrence frequency. Let $hist(SW, j)$ be the value of the histogram of SW at the j-th($1 \leq j \leq J$) bin, where J is the bin index of the maximum stroke width since each bin corresponds to a stroke width. The bin where $hist(SW)$ reaches the maximum, denoted by j^*, can be represented as

$$j^* = \arg \max_{1 \leq j \leq J} (hist(SW, j)). \tag{4.2.10}$$

We then introduce η_{sw} to describe the uniformity of stroke width within an MSER, which is defined as the ratio of contour pixels having the dominant stroke width among all contour pixels.

$$\eta_{sw} = \frac{hist(SW, j^*)}{N} \tag{4.2.11}$$

where N is the total number of the contour pixels of an MSER.

- **Ratio of Dominant Half Stroke Width**. As what was stated in subsection 4.2.1, the points of the skeleton of a character MSER usually lie along the center line of its strokes. Therefore, the distance from a character contour point to skeleton is nearly half of the length of stroke width. This distance is named as "half stroke width". Half stroke width is calculated in a different manner from calculating stroke width. A ray starts from a contour pixel along the gradient direction until a skeleton point is reached. Similar to stroke width, there is also a value of half stroke width appears most for a character MSER. Let $HSW = \{hsw^{(k)}|1, \cdots, K\}$ represent the collection of all half stroke width values of an MSER and $hist(HSW)$ be the histogram of HSW. Let $hist(HSW, k)$ be the value of the histogram of HSW at the k-th($1 \leq k \leq K$) bin, where K is maximum value of half stroke width of a character. The bin where $hist(HSW)$ reaches the maximum, denoted by k^*, can be represented as

$$k^* = \arg \max_{1 \leq k \leq K} (hist(HSW, k)). \tag{4.2.12}$$

The ratio of the number of contour points having the main half stroke width over the total number of contour points, denoted by η_{hsw}, is defined as

$$\eta_{hsw} = \frac{hist(HSW, k^*)}{N} \tag{4.2.13}$$

where N is the total number of the contour pixels of an MSER.

4.2.3 Histogram of Stroke Contour Point Gradient Direction

Character strokes tend to have double edges of opposite gradient directions and an example is shown in Figure 4.5(a). The gradient directions at the contour points of each MSER are computed. The gradient direction is quantised into eight directions. Therefore, the numbers of contour points with two opposite quantised gradient directions are approximately equal. The quantised gradient directions are represented by β_i $(i = 1, \cdots, 8)$ as illustrated in Figure 4.5(b). β_i and $\beta_{i+4}(i = 1, 2, 3, 4)$ are two opposite directions as a sibling point pair of a character MSER have approximately opposite gradient directions, which fall into the quantised gradient directions of β_i and β_{i+4}. Due to the fact that the majority of the contour points can constitute a sibling point pair with another contour point, the numbers of contour points having quantised gradient directions of β_i and β_{i+4} are close. In the histogram formed by the quantised gradient directions of the contour points of a character MSER, the heights of the bins of β_i and β_{i+4} are similar.

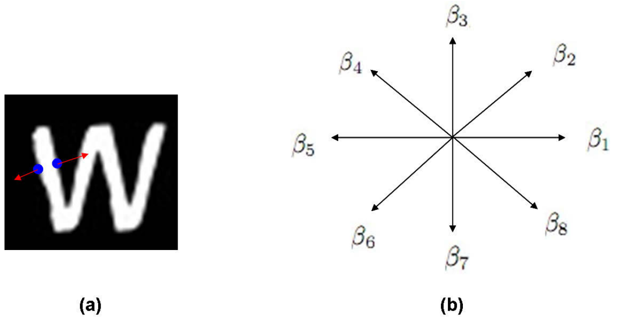


Figure 4.5: Stroke contour point gradient direction. (a) A pair of stroke edge points with opposite gradient directions. (b) Quantised gradient directions.

4.2.4 Variance of Local Foreground/Background Colour Difference (VLFBCD)

In addition to geometry-based, gradient-based and stroke-based features, colour is also utilised as an important feature in depicting a character MSER. Since scene texts usually have clear colour contrast between foreground and background, and the contrast tends to be uniform along the contour of the character. The points situated on the local neighbourhood of contour points have the most distinctive colour variation. For a contour point P of a character MSER, we consider the point P_1 locating n-pixel away from P on its gradient direction and the point P_2 locating n-pixel away from P along anti-gradient direction. The Local Foreground/Background Colour Difference (LFBCD) at P is defined as the colour difference between P_1 and P_2. As the LFBCD at each contour point of a character MSER tends to be stable, the variance of LFBCD should be low which is defined in Equation 4.2.14.

$$\sigma_{col} = \frac{1}{N}\sum_{i=1}^{N}(CD_i - \frac{1}{N}\sum_{i=1}^{N}CD_i)^2, \tag{4.2.14}$$

where

$$CD_i = |colour_i^1 - colour_i^2|. \tag{4.2.15}$$

In Equation 4.2.14, CD_i is the LFBCD at contour point i and N is the number of contour points of the MSER. In Equation 4.2.15, $colour_i^1$ and $colour_i^2$ are the colours of P_1 and P_2. The Euclidean distance is employed for computing the colour distance of CD_i in RGB colour channel.

The geometry-based, stroke-based, gradient-based and colour-based features mentioned above construct a 12-dimensional feature vector which is extracted for every MSER. A classifier trained by these features are used for discriminating text MSER from non-text MSER. Details of this classifier is given in Section 4.6. Figure 4.6 shows some examples of the character/non-character MSER classification by the classifier.

4.3 Text MSER Retrieval

The MSER classifier trained by the features discussed in the previous section can remove majority of non-character MSERs, which means the MSER classifier is good at suppressing true negatives. However, some character MSERs may also be erroneously classified as non-character MSERs. Character MSERs having very high aspect ratio, such as “l”,“i” and “I”, or having very large width strokes can easily be misclassified. Moreover, small character MSERs belonging to a word may be very close to each other and they may merge into one character MSER. In this case, the properties of character MSERs may not persist, so they are liable to be pruned in the character MSER classification step. As a result, the

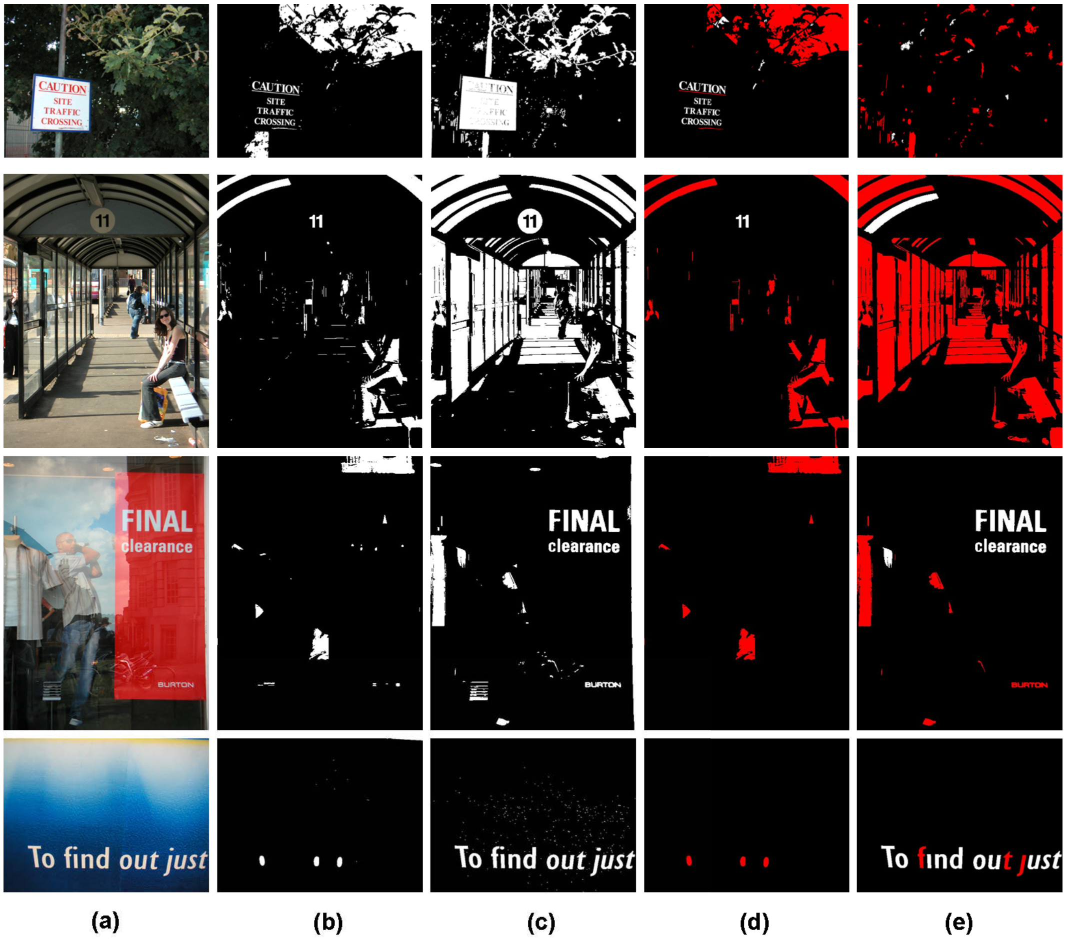


Figure 4.6: Character/non-character MSER classification. (a) Original scene images. (b) Dark-on-bright MSERs. (c) Bright-on-dark MSERs. (d) and (e) are the MSER classification results in (b) and (c) respectively. The MSERs that are classified as character are marked in white colour and the MSERs that are classified as non-character are marked in red. Best viewed in colour.

subsequent text line grouping would produce uncomplete detection bounding boxes for those text lines with some missed characters. Recovering the wrongly removed character MSERs is beneficial to enhance the performance of our algorithm. An MSER retrieval procedure is employed to achieve this goal.

In our work, MSER retrieval consists of two levels: character MSER retrieval and text line MSER retrieval. Character MSER retrieval is to call back the misclassified single character MSERs by considering the spatial vicinity, stroke width and colour of its neighbour that are classified as character MSERs. Text line MSER retrieval refers to the calling back of the small multiple character MSERs by a classifier trained by texture features.

4.3.1 Character MSER Retrieval

When considering the relationship between the correctly classified character MSERs and the misclassified character MSERs, several constraints are applied for retrieval and detailed explanations of using these constraints will be given one by one.

$$dist(M_{nr} - M_r) < 2 \times min(max(w_{nr}, h_{nr}), max(w_r, h_r)), \tag{4.3.1}$$

where $dist(M_{nr} - M_r)$ is the distance of the centroid of a non-removed MSER M_{nr} and that of a removed MSER M_r respectively. w_{nr} and h_{nr} are the width and height of the removed MSER, and w_r and h_r are the width and height of the removed MSER. This condition is for retrieving a misclassified MSER which is horizontally close to a non-removed MSER.

$$B_{nr} > \frac{T_r + B_r}{2} \text{ and } B_r > \frac{T_{nr} + B_{nr}}{2}, \tag{4.3.2}$$

where T_{nr} and B_{nr} are the top line and the bottom line of a non-removed MSER respectively, and T_r and B_r are the top and the bottom of a removed MSER respectively. This limitation is designed to retrieve the MSERs that are aligned horizontally.

$$\frac{min(h_{nr}, h_r)}{max(h_{nr}, h_r)} > Th_{height}, \tag{4.3.3}$$

where h_{nr} and h_r are the heights of a non-removed MSER and a removed MSER respectively. The heights of characters in a text line should not vary too much. Th_{height} is assigned to be 0.5.

$$dist(M_{nr}, M_r) < Th_{colour}, \tag{4.3.4}$$

where $dist(M_{nr}, M_r)$ is the difference between two colour vectors. This definition implies that the difference between the colour components of the non-removed MSER and the non-removed MSER should be less than Th_{colour}. Th_{colour} is set to be 25 as this is a suitable choice empirically.

$$\frac{max(aver\_hsw_{nr}, aver\_hsw_r)}{min(aver\_hsw_{rn}, aver\_hsw_r)} < Th_{hsw}, \tag{4.3.5}$$

where $aver\_hsw_{nr}$ and $aver\_hsw_r$ are the average half stroke width of non-removed MSER and removed MSER respectively. Due to the fact that the characters in a word have similar half stroke width, this restriction makes the retrieved MSER have similar stroke width information of its neighbour non-removed MSER. Th_{hsw} is set to 2 empirically as it is the optimal choice according to the experiment.

If a MSER removed during MSER classification satisfies all of the above constraints with its neighbouring non-removed MSER, it is retrieved as a character MSER. The retrieved MSERs will be used for retrieving other removed MSERs. This procedure terminates when no more removed MSERs can be retrieved. In Figure 4.7, we show an example of calling back misclassified character MSERs from the non-removed character MSERs.

4.3.2 Text Line MSER Retrieval

In natural scene images, the gaps between the characters of a text line are normally wide enough from each other. Therefore, the character MSERs generated from individual character in the image do not touch each other, especially for those characters that have high

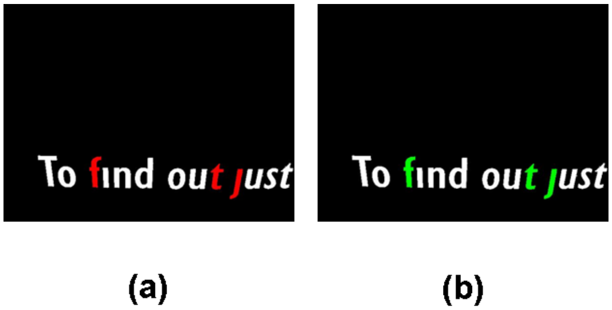


Figure 4.7: Single character MSER retrieval. (a) Classification result of MSERs where the MSERs classified as non-character are marked in red. (b) Single character MSER retrieval of (a). The retrieved MSERs are marked in green. Best viewed in colour.

contrast on the background. On the other hand, when characters are too close to their neighbouring characters, multiple character MSERs may connect together to form into one MSER. In particular, if an MSER composed of multiple characters is too small, the stroke width information calculated from it may be misleading and similar to that of a complex non-character MSER. Under these circumstances, such MSERs are highly possible to be misclassified as non-character MSERs by the single character MSER classifier and cannot be retrieved by the single character MSER retrieval scheme.

Based on the above analysis, it is necessary to propose another retrieval strategy to call back the multiple character MSERs. Compared with the single character MSERs which have salient stroke-based characteristics, a multiple character MSER holds prominent texture-based text line features at the original regions enclosed by its bounding box. Therefore, we adopt a text line classifier to retrieve multiple character MSERs. Note that what the classifier verifies is not the multiple character MSERs, but the sub-image of the original natural scene image enclosed by the bounding box of the MSER region. We call this sub-image a “counterpart text line candidate”. As a counterpart text line typically has larger width/height ratio, only the removed MSERs with a width/height greater than a threshold are considered for text line retrieval. The counterpart text lines are verified by a texture-based classifier which is presented in Section 4.5. If a counterpart text line is classified as a true text line, its MSER is labelled as a text MSER for the subsequent processing.

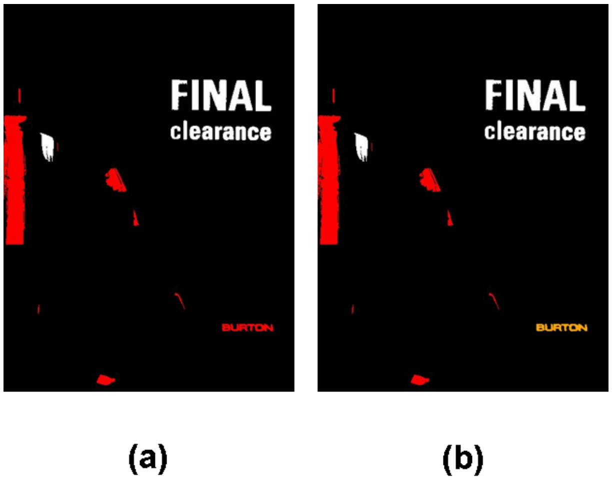


Figure 4.8: Text line MSER retrieval. (a) Classification result of MSERs where the non-character MSERs are marked in red. (b) Text line MSER retrieval of (a). The retrieved text line MSER is marked in orange. Best viewed in colour.

4.4 Character MSER Grouping

After the processing of the previous steps, the remaining MSERs need to be further merged into text lines. Characters in a text line share some similar properties, for example, height, aspect ratio, stroke width, colour and inter-character spacing. In this step, we use several rules for validating two character MSERs, MSER$_i$ and MSER$_j$, to decide whether they are to be grouped together and labelled as in one text line. The grouping process is conducted until no more MSERs can be merged. The detailed information of the five grouping rules are listed in the following equations.

Two MSERs should be labelled as in the same text line only when they are horizontally close enough to each other, i.e., they satisfy Equation 4.4.1.

$$min(abs(R_i - L_j), abs(L_i - R_j)) < 1.5 \times max(w_i, w_j), \tag{4.4.1}$$

where R_i and L_i are the maximum column and the minimum column of MSER$_i$, and R_j and L_j are the maximum column and the minimum column of MSER$_j$. w_i and w_j are the width of MSER$_i$ and MSER$_j$ respectively.

Two vertically close character MSERs belonging to different text lines should not be grouped into one text line, i.e. they satisfy Equation 4.4.2.

$$B_i > T_j \text{ and } B_j > T_i, \tag{4.4.2}$$

where T_i is the top of MSER$_i$, B_i is the bottom of MSER$_i$, T_j is the top of MSER$_j$ and B_j is the bottom of MSER$_j$.

The heights of the characters of a text line normally do not vary much.

$$\frac{min(h_i, h_j)}{max(h_i, h_j)} > Group\_Thresh_{height}, \tag{4.4.3}$$

where h_i and h_j are the heights of $MSER_i$ and $MSER_j$ respectively. The value of $Group\_Thresh_{height}$ is set to be 0.5 in the experiment.

The colours of characters in a text line are typically uniform, so the colour difference should be within a range.

$$Dis(C_i, C_j) < Group\_Thresh_{colour}, \tag{4.4.4}$$

where $Dis(C_i, C_j)$ is the colour difference between $MSER_i$ and $MSER_j$. $Group\_Thresh_{colour}$ is 25 in the experiment.

Characters in one text line should be have close half stroke width.

$$\frac{max(aver\_hsw_i, aver\_hsw_j)}{min(aver\_hsw_i, aver\_hsw_j)} < Thresh_{hsw}, \tag{4.4.5}$$

where $aver\_hsw_i$ and $aver\_hsw_j$ are the half stroke widths $MSER_i$ and $MSER_j$ respectively. $Thresh_{hsw}$ is 2 in the experiment.

In fact, only single character MSERs need to be grouped into text lines as text rarely appears in the form of single characters. Based on the multiple character MSER retrieval processing in the last section, the MSERs fed to MSER grouping may contain retrieved multiple character MSERs which should not be considered in the grouping step. To discriminate single character MSERs and multiple character MSERs, the width/height ratio of an MSER is used as the criterion. If the width/height ratio of an MSER is less than a threshold, it is used for MSER grouping. Otherwise, it is treated as a multiple character MSER and the grouping step is skipped. The single character MSERs grouped into different text lines are enclosed by bounding boxes as text line candidates. All text line candidates are further verified as text lines or non-text lines in the following false alarm elimination processing.

4.5 False Alarm Elimination

Complex structures, such as leaves, bricks, windows, fences and some other structures similar to the stroke of text may exist in the detection results. In order to eliminate these false alarms, a trained classifier is applied. The IT-LBP descriptor proposed in Chapter 3 has shown its effectiveness in text/non-text classification, so it is adopted to depict text lines for non-text elimination. The text lines classification results in both the dark-on-bright MSER map and the bright-on-dark MSER map and the remaining text lines are the final detection result.

In order to do the classification, each bounding box of a detected text line candidate is normalised into 20-pixel high with the original aspect ratio. A scanning window with size of 20-by-20 moves along the bounding box and the IT-LBP is calculated with the scanning window. The final decision score is a weighted sum of all decision scores of the regions enclosed by the scanning window.

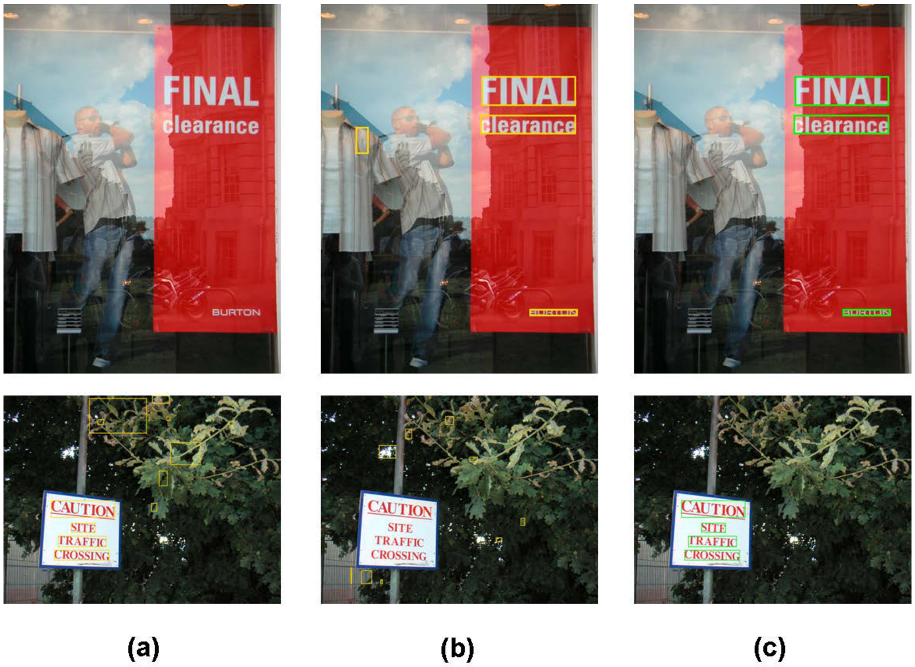


Figure 4.9: Text line candidates false alarm elimination. (a) Text line candidates (enclosed by yellow bounding boxes) obtained from dark-on-bright MSER map. (b) Text line candidates (enclosed by yellow bounding boxes) obtained from bright-on-dark MSER map. (c) Final detected text lines (enclosed by green bounding boxes) after false alarm elimination. Best viewed in colour.

Due to the limited number of training samples, the trained text line classifier may generate some false positive detection in the final detection results. In order to further improve the classification capability of the classifier, a "bootstrap" classifier training scheme is applied as in [7,123] to improve the performance of the classifier. The false positive detection results will be supplemented into the training set as negative training samples and the classifier will be trained again with the added negative samples. The bootstrap classifier training procedure is demonstrated in Figure 4.10.

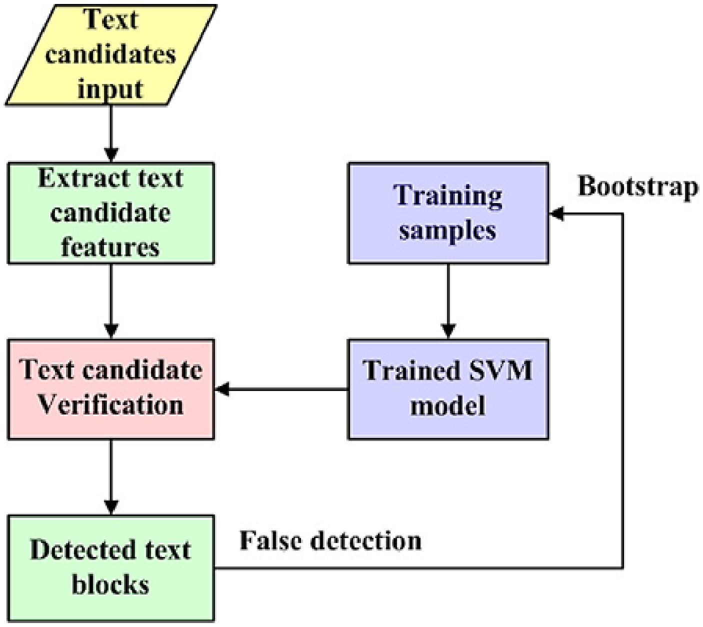


Figure 4.10: The procedure of bootstrap classifier training (courtesy Wei [7]).

4.6 Experimental Results

In order to evaluate the performance of our method, the Robust Reading Competition Challenge 2: Reading Text in Scene Image dataset released for ICDAR2011 is used as a natural scene text detection benchmark dataset. There are 229 images for training and 255 image for testing contained in this dataset. The sizes of images range from 422×102 to 3888×2592.

There are two classifiers embedded in our system: one is for character/non-character MSER classification and the other one is for text/non-text line classification. The positive and negative samples used for training these two classifiers are from the ICDAR2011 Robust Reading Competition Challenge 2 dataset, which includes a group of images for Text Localization Task and a group of images for Word Recognition Task. For training

Feature Set 1	Feature Set 2	Feature Set 3	Feature Set 4
88.9134%	93.9169%	96.6218%	98.0729%

Table 4.1: Comparisons of classification accuracy rates of MSER classifiers using different sets of features.

the single character MSER classifier, we extract MSERs from the images in the Text Localization Task. In total, 8614 MSERs are generated in which 3403 MSERs are labelled as positive samples and 5211 MSERs are labelled as negative MSERs manually. In order to evaluate the contribution of the features in character/non-character MSER classification, the classification accuracy of classifiers trained using different combinations of different types of features are compared. All features are divided into four sets: Set 1 is geometry-based features, Set 2 is the combination of geometry-based and stroke-based features, Set 3 is the combination of geometry-based, stroke-based and gradient-based features, Set 4 is the combination of all types of features. The comparisons of classification accuracy rates on the training samples by applying different MSER classifiers with the four feature sets are illustrated in Table 4.1. As shown in Table 4.1, the classifier using all types of features achieves the best performance and it is utilised for character/non-character MSER classification.

For training the text line classifier, all of the positive training samples are obtained from the Word Recognition training dataset. The word images in this dataset are text line images cropped from the Text Localization test dataset. Although this dataset was prepared for word recognition, they satisfy the requirements of training sample creation. Each word image is normalised to 20-pixel high with the original aspect ratio, and then divided into 20×20 sub-images with 10-pixel moving step. The negative samples were obtained in the same way. These 20×20 sub-images are used as training samples. There are 5188 positive samples and 5500 negative samples for training. The Vedaldi's algorithm [124] is implemented for extracting MSERs and the two classifiers are trained by using the LIBSVM package [118]. Some natural scene text detection results using the ICDAR2011 Robust Reading Competition Challenge 2 dataset are demonstrated in Figure 4.11.

Three criteria, recall, precision and harmonic means (or f-measures) as the same measurements in the ICDAR2011 Reading Text in Scene Images Competition [2], are utilised to quantitatively compare our method with other existing methods. The definitions are given in Equation 4.6.1, Equation 4.6.2 and Equation 4.6.3. Table 4.2 lists the comparisons of different algorithms assessed by the above three criteria. We implement two sets of experiments for comparison: Comparison 1 and Comparison 2. Comparison 1 is obtained by our algorithm without false alarm elimination step and Comparison 2 is obtained by our algorithm without MSER retrieval step and false alarm elimination step.

$$\text{Precision=}\sum_{r_e \in E} m(r_e, T)/|E|, \tag{4.6.1}$$

$$\text{Recall=}\sum_{r_t \in T} m(r_t, E)/|T|, \tag{4.6.2}$$

$$\text{Harmonic Mean=}\frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{4.6.3}$$

where $m(r, R)$ is the best match for a rectangle r in a set of rectangles. R, E and T are the estimated and ground truth rectangles respectively.

The experimental results have demonstrated the good performance of the proposed algorithm in both of the qualitative and quantitative aspects. It can be seen from Figure 4.11 that the proposed algorithm has quite few false alarms and most of the text lines can be detected. The ranking of the algorithms is based on the harmonic means in Table 4.2. The

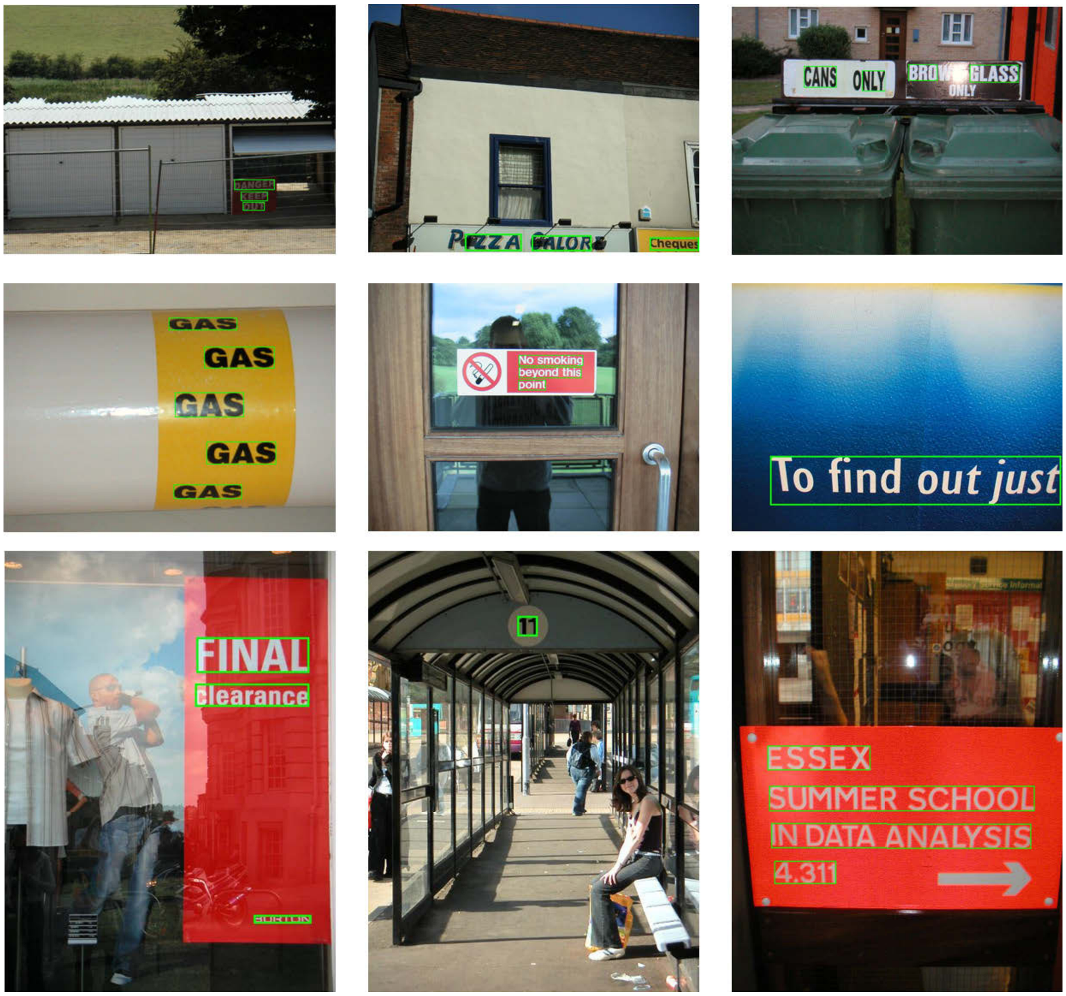


Figure 4.11: Some scene text detection results using ICDAR2011 Robust Reading Competition Challenge 2 dataset by our algorithm.

higher the harmonic mean is, the better performance of an algorithm has. In the comparison within our algorithm, the harmonic mean is only 60.62% without the character MSER retrieval step and the false alarm elimination step. With the character MSER retrieval step, our algorithm can achieve a higher harmonic mean of 63.57%. The main reason is the improvement of the recall rate. When the false alarm elimination step is also performed, the harmonic mean can reach up to 80.46%, which is the highest among all algorithms under

Method	Recall	Precision	Harmonic Mean
Our algorithm	**71.91%**	**91.33%**	**80.46%**
Shi's method [22]	63.1%	83.3%	71.8%
Kim's method [2]	62.47%	82.98%	71.28%
Our algorithm (Comparison 1)	**71.83%**	**57.01%**	**63.57%**
Yi's method [2]	58.09%	67.22%	62.32%
TH-TextLoc System [2]	57.68%	66.97%	61.98%
Our algorithm (Comparison 2)	**61.99%**	**59.30%**	**60.62%**
Neumann's method [2]	52.54%	68.93%	59.63%
TDM IACS [2]	53.52%	63.52%	58.09%
LIP6-Retin [2]	50.07%	62.97%	55.78%
KAIST AIPR system [2]	44.57%	59.67%	51.03%
ECNU-CCG method [2]	38.32%	35.01%	36.59%
Text hunter [2]	25.96%	50.05%	34.19%

Table 4.2: Performance comparisons between our method and some state-of-the-art algorithms using the ICDAR2011 Robust Reading Competition Challenge 2 dataset [2].

comparison. This means that the false alarm elimination step greatly enhance the precision of the proposed algorithm.

4.7 Summary

This chapter presents an algorithm designed for detecting text in scene images. Based on the characteristics of uniform intensity and strong contrast against background that scene text have, Maximally Stable Extremal Regions (MSER) is used to generate the connected components of character strokes of text. Geometry-based, stroke-based, gradient-based and colour-based features are applied to depict character MSERs and a single character MSER classifier is trained for removing the non-character MSERs. Two schemes are applied separately to retrieve the misclassified single character MSERs and multiple character MSERs sequentially to compensate the false negatives caused by the character MSER classifier. All obtained character MSERs are aggregated into text lines in the grouping process. Finally, false positives are suppressed by a classifier trained with a texture-based text feature. The experimental result results have shown that the proposed method outperforms all state-of-the-art methods involved in the ICDAR2011 Robust Reading Competition Challenge 2. The character MSER retrieval step contributes to the high recall rate and the false alarm elimination step contributes to the high precision rate.

Chapter 5

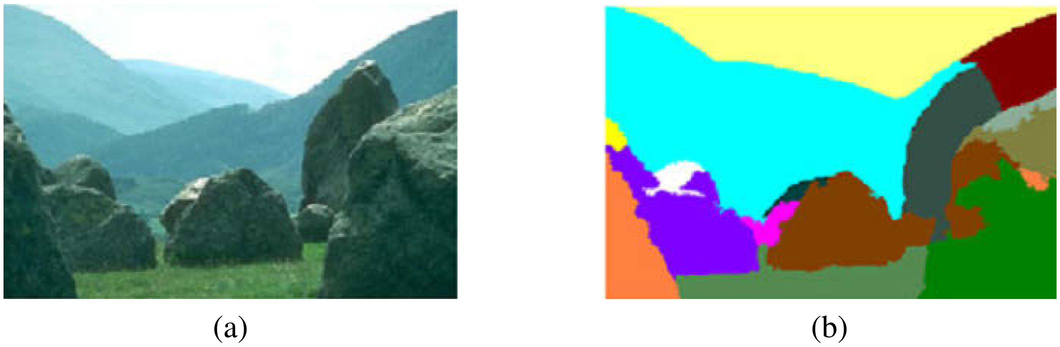
Text Binarisation

In Chapter 3 and Chapter 4, it has been shown that text detection can be used to find text lines in an image and that different bounding boxes can be used to enclose the local areas of individual text lines. The next stage of concern is how the characters within each enclosed text lines may be correctly recognised. As the text line within a bounding box is a sub-image of the original image which contains only text, it is called a text image in this thesis. In other words, one requires to find out the characters in the extracted text images. This task is accomplished by character recognition which converts the characters in text images into ASCII codes as the recognition results of the characters.

Optical character recognition (OCR) is a technique that conducts the mechanical or electronic conversion from the scanned document of printed and typewritten text into machine-encoded text. OCR has been very successful on recognising characters in good quality typed documents and been widely applied to practical text-reading applications. However, when suffering from degradations, such as poor contrast, uneven lighting, blur, complex background and so on, the recognition result could be badly influenced if a text image is sent to an OCR engine without further processing.

In order to increase the recognition rate, text images usually need to be converted to binary images before being sent to an OCR system for recognition. This conversion processing is called text binarisation which means a text image is transferred into a binary image with the same size of the original text image, and white pixels represent text and black pixels represent background. The accuracy of text binarisation can significantly affect the text recognition rate. Scene text is a kind of text that easily suffers from uneven lighting and complex background which is a challenging text binarisation process. In order to enhance the performance of the subsequent text recognition, a gray level-based algorithm and a colour-based text binarisation algorithm are proposed in Section 5.1 and Section 5.2 respectively. Experimental results of the proposed gray level-based method and the proposed colour-based method are presented and compared with other methods in Section 5.3. Summary is presented in Section 5.4.

In Section 5.1, a gray level-based technique is discussed in order to motivate the proposed method. A concise rationale is given in Section 5.1.1. Section 5.1.2 provides the theoretical background of the mean-shift algorithm. This is followed by technical details of colour channel selection by mean-shift in Section 5.1.3. Since image segmentation is a technique to separate pixels of an image into multiple categories, text binarisation is an image segmentation problem which fixes the number of the classified categories into two. As reviewed in Chapter 2, the graph-based image segmentation approaches are reliable and have become a thriving research area. Therefore, the graph-based image segmentation technique is explored in Section 5.1.4 for binarising the selected channel image.



(a) (b)

Figure 5.1: An example of image segmentation. (a) The original image. (b) The segmentation result of (a). Sub-regions are represented by different colours (courtesy Zhang [8]).

5.1 Gray level-Based Text Binarisation

In a text information extraction system, text binarisation is to separate the pixels of a text image into text as foreground and anything else as background. The foreground is composed of the pixels belonging to the characters in the text image, and the background is composed of the pixels belonging to the non-character regions. In this sense, text binarisation is a particular case of image segmentation in the application of text information extraction. Image segmentation is a significant step in many practical applications of image analysis and computer vision. The target of image segmentation is to partition an entire digital image into multiple non-overlapping sub-regions of which pixels contain certain common properties. Each segmented sub-region is marked using a specific label assigned to all of its pixels to differ from other sub-regions. An example of image segmentation is given in Figure 5.1. “sky”, “mountain”, “stones” and “lawn” in the image are separated into individual objects and different colours are used as labels to stand for different objects. The segmented multiple regions of interest are meaningful for further processing. A specific case of image segmentation, which is called binary image segmentation, is to divide all of the pixels in an image into two groups: foreground and background. Foreground is the whole of the regions-of-interest of the original image and background is the whole of the remaining parts. The two typical colours used for foreground and background are black and white respectively. A myriad of techniques on image segmentation have been reported in literatures [125] and [8].

Gray level-based binarisation methods seek for an optimal thresholding gray level image to separate an image into text and background. The pixels with gray levels greater (or lower) than the threshold value are classified as foreground (or background) pixels. Otsu's method [78] is a well-known image segmentation technique which is used to perform image thresholding based on histogram, under the assumption that the image to be thresholded has bi-modal histogram. Otsu's method selects the threshold by minimising the within-class variance of the two groups of pixels. A histogram is composed of bins and the height of each bin is determined by the number of pixels with certain intensity value or the occurrence frequency of a certain intensity if the histogram is normalised. The idea of classifying the pixels of an image into two classes with least differences among the pixels of each class corresponds to the task of text binarisation. In the senario of light text on the dark background or dark text on the light background, a large number of pixels are of relatively large intensity values and a large number of pixels are of relatively low intensity values. The bins in the histogram distribute primarily around two intensity values far away from each other. The two main peaks imply the dense occurrences of text pixels and background pixels around the two well-separated intensity values. Taking the histogram in Figure 5.2 as an example, one main peak is at 70 and another main peak is at 220.

Although the shape of the histogram of a gray level image can provide a clue for text binarisation, a good binarisation result may not be obtainable by setting a certain value to threshold the histogram if the distribution of the histogram is not clearly bi-modal. The

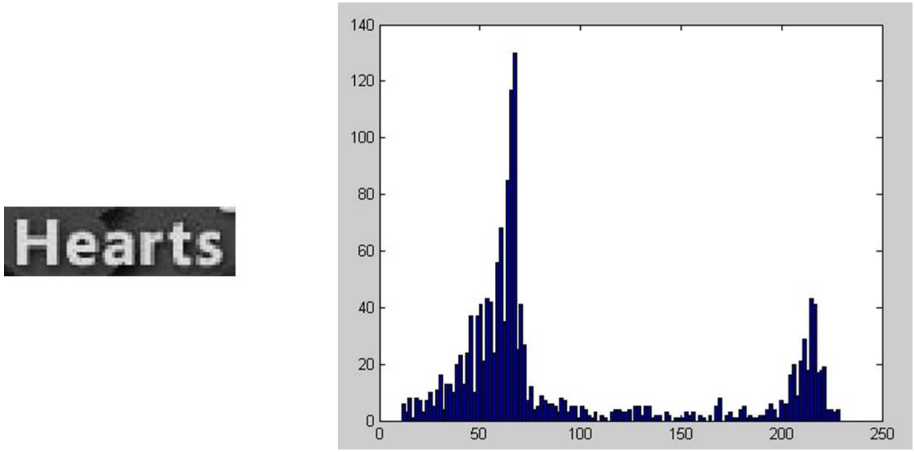


Figure 5.2: An example of the histogram of a gray level text image. The figure on the right is the histogram of the gray level image on the left. The range of intensity values of a gray level is from 0 to 255. The horizontal axis of the histogram is the intensity values. The vertical axis of the histogram is the number of pixels belonging to a certain intensity value.

generation of a gray level image is by the conversion from colour channels to the shades of gray varying from black as the weakest to white as the strongest. The conversion is a linear combination of colour channels. However, this may result in the loss of the important visual clues presented from different colour channels. The colour of text may be quite different from that of background, but after colour/gray conversion, their gray level intensities may be quite similar as many different colours can have the same gray level value. As the instance illustrated in Figure 5.3, (0,170,0), (230,53,0), (80,83,240) and (230,14,200) are four totally different colours in RGB colour channel and they have an unique gray level value of 100 following the conversion in Equation 5.1.1. Thus, the histogram of the gray level image will not show the bi-modal shape and this will be very difficult to get a good segmentation through histogram thresholding.

$$\text{Gray level intensity} = 0.299R + 0.587G + 0.114B \tag{5.1.1}$$

where R, G and B are the channel components in RGB colour model and they are integers between 0 and 255. The gray level intensity is the rounded value of the sum in the right hand side.

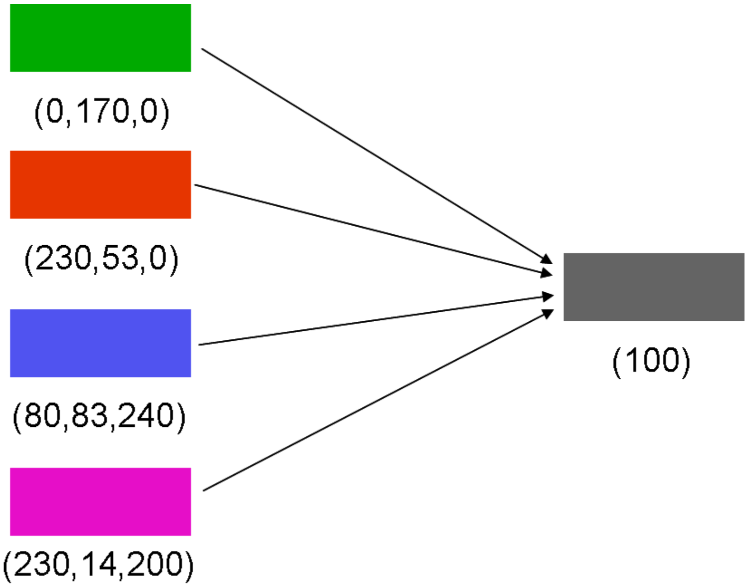
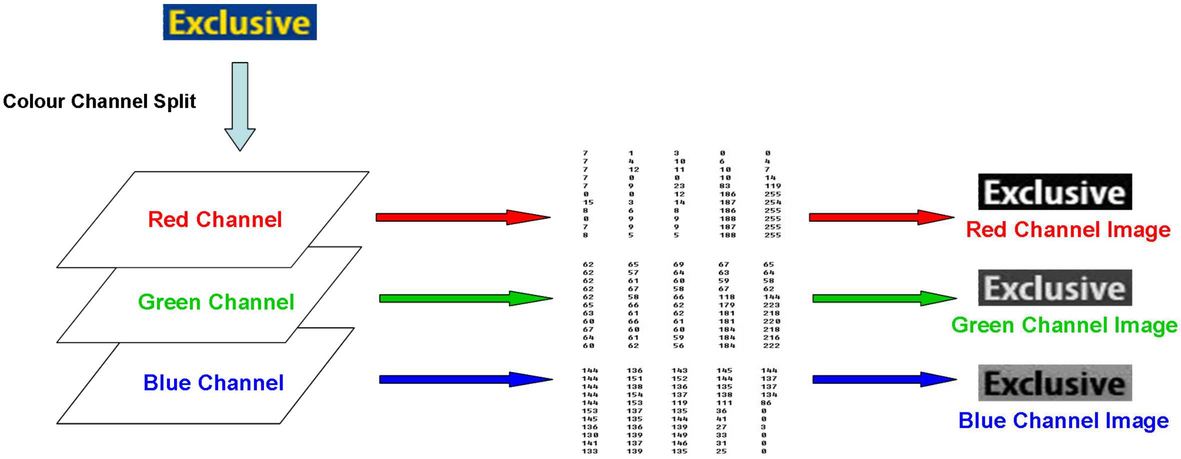


Figure 5.3: Different colours can be converted into an identical gray level value.



7	1	3	0	0
7	4	10	6	4
7	12	11	10	7
7	0	0	10	14
7	9	23	83	119
0	0	12	186	255
15	3	14	187	254
8	6	8	186	255
0	9	9	188	255
7	9	9	187	255
8	5	5	188	255

62	65	69	67	65
62	57	64	63	64
62	61	60	59	58
62	67	58	67	62
62	58	66	118	144
65	66	62	179	223
63	61	62	181	218
60	66	61	181	220
67	60	60	184	218
64	61	59	184	216
60	62	56	184	222

144	136	143	145	144
144	151	152	144	137
144	138	136	135	137
144	154	137	138	134
144	153	119	111	86
153	137	135	36	0
145	135	144	41	0
136	136	139	27	3
130	139	149	33	0
141	137	146	31	0
133	139	135	25	0

Figure 5.4: Colour channel split on RGB colour space.

In order to avoid the loss of contrast between text and background when converting a colour image to a gray level image, we explore a novel binarisation method based on a colour channel. The main colour component of text normally has variation to that of its background. Therefore, the distributions of text colour and background colour are different in the same colour channel. Colour channels are colour components that constitute a colour image. In Figure 5.4, the red, green and blue components of a colour image are split into three colour channels. Each colour channel consists of colour component strengths at all pixels. Since the range of colour component strength is from 0 to 255, a colour channel can be treated as a gray level image with the same size of the colour image and the value at each pixel stands for the intensity of the corresponding colour component. The gray level image is referred to as a colour channel image. The two main peaks in the histogram of a colour channel image imply the distribution of the text intensity and background intensity in the corresponding colour channel.

The histogram of the colour channel image having the wider separation of the two main peaks means the greater inter-class intensity difference of text and background. Therefore, performing text binarisation on the colour channel image with the widest bi-modal distance can avoid misclassification of text pixels and background pixels. Similar ideas have been reported in the literature of single character image binarisation [67, 68]. Under the assumption that the wider the histogram of grayscale image is, the easier it is to binarise the image with a threshold value, Yokobayashi's method [67] selected the colour channel with the maximum breadth of histogram in the Cyan/Magenta/Yellow colour space. A local 3×3 neighbourhood at each pixel was considered to generate the binarised image. If five or more pixels of the total pixels in the neighbourhood have smaller gray levels than the mean of the gray levels of the whole image, the pixel was set as a foreground (character) pixel. Otherwise, the pixel was set as a background pixel. Both the positive and negative binarisation results are used as input for recognition. In [68], all colour points were projected onto a total of 180×180 axes using spherical polar coordinates $(\gamma, \theta, \varphi)$ in the RGB colour space first. Then, the Otsu's binarisation technique was applied to compute a maximum between-class separability by setting an optimal threshold value for each axis. Finally, the axis having the largest between-class separability was selected and the corresponding threshold was used for binarising the input image. Nevertheless, neither of these two methods can be applied for binarising text images containing text lines that have multiple characters. Because these methods aimed at binarising single character images. A text line with multiple characters need to be further divided into single characters by the process of character isolation by using the two mentioned methods. In this research, text binarsation is performed without character isolation.

5.1.1 Rationale

Red, green and blue are three primary additive colours that are added together to form a desired colour as individual components. We select the channel image (among the Red, Green and Blue channel images) whose histogram has the biggest distance between its two main peaks. The reason is that the colour component in the selected channel image reflects the greatest foreground/background inter-class difference. In the next step, a graph is constructed based on the selected channel image to depict the similarity of intra-class pixels and difference of inter-class pixels. Finally, in order to get an optimal binarisation result, normalized cut is used to cut the graph into two sub-graphs which represent the foreground (i.e., the text) and the background respectively.

The task of text binarisation is to separate the foreground (text) from the background in a text image. In a text image without degradation, the two main peaks representing the main

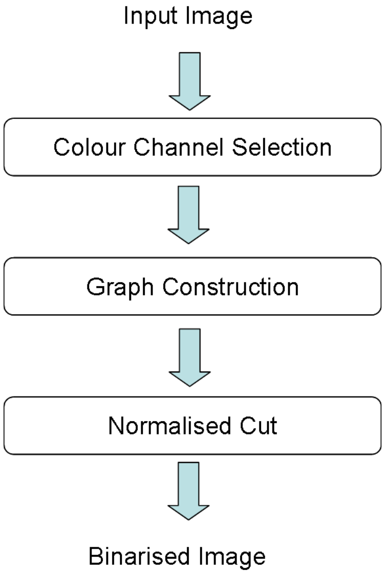


Figure 5.5: The flow chart of our colour channel image-based text binarisation method.

foreground colour and main background colour in the intensity histogram of the image are usually well apart. For such kind of clear images, some classic binarisation techniques, such as Otsu's thresholding method [78], can find the optimal threshold to binarise the images satisfactorily. However, for scene text images that are degraded by uneven lighting or with complex background, the two main peaks may not be clearly separable in most cases. Figure 5.6 shows a text image without degradation and a text image with degradation and their intensity histograms.

Similarly, like a gray level image, colour channel image of a text image may have more than two peaks in its corresponding histogram. Our target is to find the colour channel image whose histogram has two most-separative peaks. By taking the advantage of the mode seeking power of the mean-shift [126] algorithm, the main peaks in all of the three channel images can be located by seeking the positions of the two modes in their histograms. The positions of the two estimated peaks reflect the separability of the text cluster and the

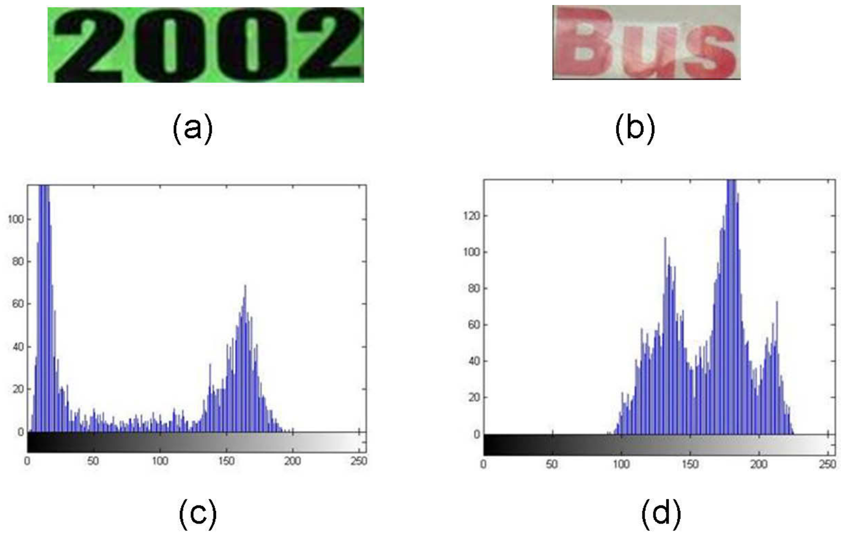


Figure 5.6: Histograms of a clear text image (a) and a degraded text image (b). Sub-figures (c) and (d) are the histograms of the gray level images of (a) and (b) respectively.

background cluster. The distance between the two peaks will be used as the criterion for evaluating the inter-cluster difference between text and background. The colour channel possessing the greatest distance between its two peaks in its histogram will be chosen for further processing.

After the desired colour channel image is obtained, the pixels in a text image will be split into two classes: text and background. The problem of text binarisation is transferred to graph segmentation in our framework. A graph is constructed and then the normalised cut algorithm is applied to divide the nodes into two groups based on the relationship of nodes. The separation of nodes corresponds to text binarisation.

5.1.2 The Mean-shift Algorithm

For the purpose of choosing the colour channel image whose histogram has two most-separative peaks, the mean-shift algorithm is used. In this subsection, the theoretical knowledge of the mean-shift algorithm is introduced. The performance in selecting channel image by using the mean-shift algorithm is given in the next subsection.

The mean-shift algorithm is a simple iterative procedure that shifts each data point to the average of data points in its neighborhood [126]. Compared with the traditional K-means clustering algorithm, mean-shift is free of setting the number of clusters at the initial stage. The mean-shift algorithm was firstly presented by Fukunaga and Hostetler [127] and was later introduced into low-level vision problems in Comaniciu and Meer's work [128]. Due to its successful applications in image segmentation [129] and object tracking [130], mean shift has become a well known mode seeking technique in computer vision.

Density Gradient Estimation

Given n data points $\boldsymbol{x}_i, i = 1, 2, \cdots, n$ on a d-dimensional space R^d, the multivariate kernel density estimation obtained with kernel $K(\boldsymbol{x})$ and window radius h is

$$\hat{f}(\boldsymbol{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right). \tag{5.1.2}$$

For radially symmetric kernels, it suffices to define the profile of the kernel $k(\boldsymbol{x})$ satisfying

$$K(\boldsymbol{x}) = c_{k,d} k(||\boldsymbol{x}||^2). \tag{5.1.3}$$

The kernel $K(\boldsymbol{x})$ satisfies

$$\begin{aligned}
&\int_{R^d} K(\boldsymbol{x})d\boldsymbol{x} = 1 \\
&\lim_{\|\boldsymbol{x}\|\to\infty} \|\boldsymbol{x}\|^d K(\boldsymbol{x}) = 0 \\
&\int_{R^d} \boldsymbol{x}K(\boldsymbol{x})d\boldsymbol{x} = 0 \\
&\int_{R^d} \boldsymbol{x}\boldsymbol{x}^T K(\boldsymbol{x})d\boldsymbol{x} = c_K\mathbf{I},
\end{aligned} \tag{5.1.4}$$

where $c_{k,d}$ is a normalisation constant.

The estimate of the density gradient is defined as the gradient of the kernel density estimate in Equation 5.1.2

$$\begin{aligned}
\hat{\nabla} f(\boldsymbol{x}) &\equiv \nabla \hat{f}(\boldsymbol{x}) \\
&= \frac{1}{nh^d}\sum_{i=1}^{n} \nabla K\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right) \\
&= \frac{2c_{k,d}}{nh^{d+2}}\sum_{i=1}^{n} (\boldsymbol{x}_i-\boldsymbol{x}) g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right) \\
&= \frac{2c_{k,d}}{nh^{d+2}}\left[\sum_{i=1}^{n} g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)\right]\left[\frac{\sum_{i=1}^{n}\boldsymbol{x}_i g\big(||\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}||^2\big)}{\sum_{i=1}^{n} g\big(||\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}||^2\big)} - \boldsymbol{x}\right],
\end{aligned} \tag{5.1.5}$$

where $g(s) = -k'(s)$. The first term

$$\frac{2c_{k,d}}{nh^{d+2}}\left[\sum_{i=1}^{n} g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)\right] \tag{5.1.6}$$

is proportional to the density estimation at $\boldsymbol{x}$ computed with kernel $G(\boldsymbol{x}) = c_{g,d}g(||\boldsymbol{x}||^2)$ and the second term

$$\mathbf{m}_h(\boldsymbol{x}) = \frac{\sum_{i=1}^{n}\boldsymbol{x}_i g\big(||\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}||^2\big)}{\sum_{i=1}^{n} g\big(||\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}||^2\big)} - \boldsymbol{x} \tag{5.1.7}$$

is the mean shift. The mean shift vector always points to the direction of the maximum increase in the density. The mean shift procedure proceeds in a successive manner:

- computation of the mean shift vector $\mathbf{m}_h(\boldsymbol{x}^t)$,

- translation of the window $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + \mathbf{m}_h(\boldsymbol{x}^t)$.

As the successive step repeats iteratively, the sequence of $\mathbf{m}_h(\boldsymbol{x}^t)$ converges and at a point where the gradient of density function is zero, $\nabla \hat{f}(\boldsymbol{x}) = 0$, is reached.

Convergence

Let $\{\boldsymbol{y}_j\}_{j=1,2,\cdots}$ denote the sequence of the center position of successive locations of kernel G,

$$\boldsymbol{y}_{j+1} = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(||\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}||^2\right)}{\sum_{i=1}^{n} g\left(||\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}||^2\right)} - \boldsymbol{x}, j = 1, 2, \cdots \tag{5.1.8}$$

is the weighted mean at $\boldsymbol{y}_j$ computed with kernel G and $\boldsymbol{y}_1$ is the center of the initial position of the kernel. The corresponding sequence of density estimates computed with kernel K, $\{\hat{f}_{h,K}(j)\}_{j=1,2,\cdots}$, is

$$\hat{f}_{h,K}(j) = \hat{f}_{h,K}(\boldsymbol{y}_j), j = 1, 2, \cdots . \tag{5.1.9}$$

It has been mathematically proved that the sequences $\{\boldsymbol{y}_j\}_{j=1,2,\cdots}$ and $\{\hat{f}_{h,K}(j)\}_{j=1,2,\cdots}$ converge [128]. The convergence of the sequences $\{\boldsymbol{y}_j\}_{j=1,2,\cdots}$ guarantees that the center position of the kernel will ultimately reach a stable point.

In this approach, the mean-shift algorithm [127] is chosen to estimate the two main peaks produced by the three intensity histograms of the images extracted from the Red, the Green and the Blue channels of an image respectively. The channel image that has two most separated main peaks in its histogram is selected for text binarisation.

5.1.3 Mean-Shift Based Channel Image Selection

The distance of the two main peaks in the histogram of a colour channel image indicates the extent of the difference between the foreground and the background of the channel image.

The greater the distance is, the more distinctive the foreground pixels and the background pixels are. According to the observation, the shapes of the histograms of different colour channel images present the different distributions of foreground pixels and background pixels. The colour channel image of the colour component that implies the greatest disparity of foreground and background has the widest distance of the two main peaks in the histogram. For example, in Figure 5.7, the histogram of the B channel image has a bigger gap between its two main peaks than the other three histograms (including the histogram shown in Figure 5.7(b2) produced from the gray-level image) have. The red lines in Figures 5.7 (b2), (c2), (d2) and (e2) indicate where the peaks are.

In order to find the two main peaks in the histogram of each channel image, the mean-shift algorithm is used. First, the density function of each channel image histogram is estimated with a kernel function, and then the mean-shift algorithm is performed to find the two local maxima, which reflect the corresponding colour component distribution of the foreground pixels and background pixels respectively in the histogram. There are two factors that should be considered when applying the mean-shift algorithm: the kernel function and the window radius. In this work, the Gaussian kernel $K(\boldsymbol{x}) = (2\pi)^{-d/2}exp(-\frac{1}{2}||\boldsymbol{x}||^2)$ is chosen as the kernel function, where d is the dimensionality of data. The window radius h is set as 1 in our experiments. Every channel image is fed into the density estimate function in Equation 5.1.2.

After the two local maxima C_{peak1} and C_{peak2} in the histogram of each channel image are found and the distance $|C_{peak1} - C_{peak2}|$ is computed, the channel image having the biggest distance is selected for text binarisation. In the example shown in Figure 5.8, it can be seen that C_{peak1} and C_{peak2} overlap in (b) and (d), and C_{peak1} and C_{peak2} are well-separated in (c). Therefore the distance between $|C_{peak1}$ and $C_{peak2}|$ in the G channel image

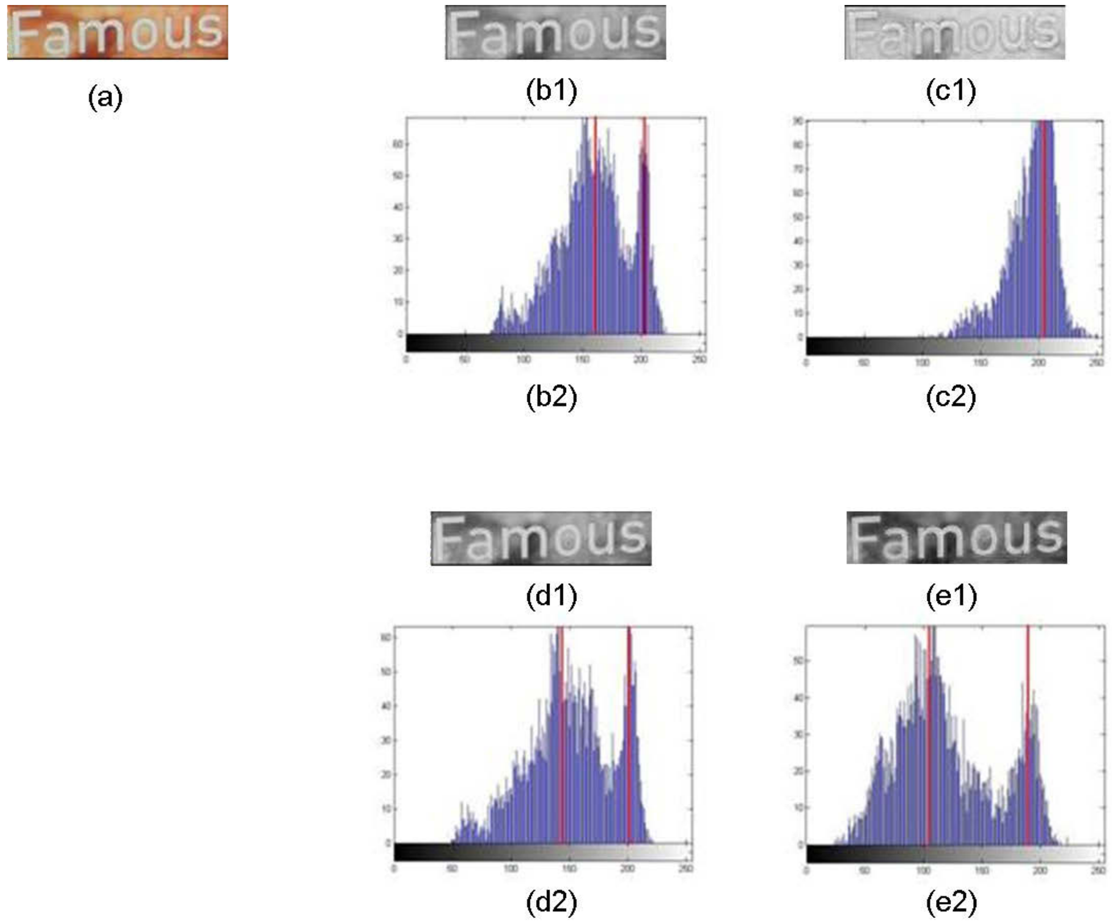
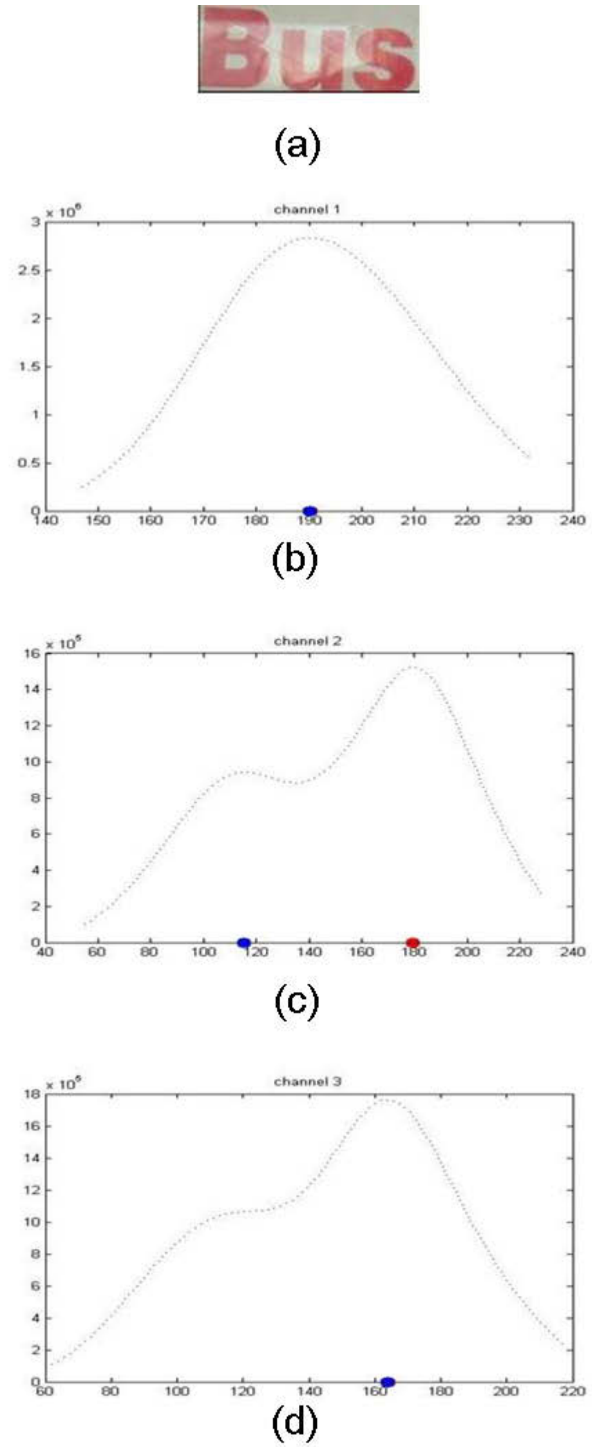


Figure 5.7: The main peaks in different histograms of a colour text image. (a) The original colour image. (b1) The intensity map of (a). (c1), (d1) and (e1) are the images of R channel, G channel and B channel of (a) respectively. (b2), (c2), (d2) and (e2) are the histograms of (b1), (c1), (d1) and (e1) respectively. The red lines in each histogram indicates the locations of peaks (Best view in colour).



(a)

(b)

(c)

(d)

Figure 5.8: The two located local maxima in the histograms of the R, G and B channel images for a sample image (a). The curves shown in (b), (c) and (d) display the estimated density distributions (i.e. histograms) of R, G and B channel images respectively. The blue and red spots (best viewed in colour) are the positions of the two intensity values C_{peak1} and C_{peak2} (indicating the peaks of each histogram).

is bigger than in the R and B channel images. Thus, for the image shown in Figure 5.8 (a), the G channel image is selected for text binarisation.

5.1.4 Graph-Based Selected Channel Image Segmentation

Otsu's method may be directly applied to the selected colour channel image directly for text binarisation. However, Otsu's method only considers pixel value to get the desired optimal threshold value. Apart from pixels values, the spatial locations of the pixels are also taken into account and graph is employed to accomplish text binarisation. A graph is a representation of a set of nodes and the linkage between different nodes is determined by predefined characteristics. To separate the text pixels from the background pixels, we formulate the problem in graph segmentation framework. A group of pixels having a same intensity value of the selected channel image is defined as "node" and the difference between two groups of pixels is represented by "edge". Then, the optimal segmentation of foreground pixels and background pixels are performed on the graph.

Next, basic concepts and notations of graph theory are given before the definitions of graph construction and techniques of bi-separating the graph into text and background.

Graph Construction

Mathematically, a graph $G = (V, E)$ is composed of a set of nodes (or vertices) $v \in V$ and a set of edges $e \in E \subseteq V \times V$ connecting two nodes $v_i \in V$ and $v_j \in V$. There are two types of edges: undirected edge and directed edge. An undirected edge is an unordered pair of nodes $e_{\{v_i,v_j\}}$. The edges of an undirected graph do not have orientation which means that there is no symmetric relation of nodes. A directed edge is an ordered pair of nodes $e_{\{v_i,v_j\}}$, in which v_i is called the starting node and v_j is called the ending node. The weight of an edge is a value assigned to the edge, which describes the strength of relationship between the two nodes. In graph-based image segmentation models, a node can be a pixel, a set of pixels with common characteristics, a super pixel, or a feature vector of the pixel. The edge weight w is defined to describe the similarity or dissimilarity of two nodes connected by it according to the specific application.

From the analysis of the histogram of the selected channel image in Section 5.1, we can notice that the majority of pixel values are densely distributed near either of the two well-separated pixel values. The reason is that the pixels belonging to the same class (text or background) have similar values. Therefore, the pixel values play an important role in depicting the inter-class pixels and intra-class pixels. Meanwhile, except the boundary of text and background, the pixels in the same class are spatially close to each other. For instance, text pixels typically appear in the neighbourhood of a text pixel and those pixels have similar values. In one word, the value of a pixel and the pixel's local neighbourhood where similar pixel values exist are crucial in depicting the relationship among pixels.

Li et al. [75] provided a text binarisation algorithm to segment a text line with multiple text colours. The original colour text image was converted into a gray-level image which was further divided into multiple binary sub-images via recursive graph bi-partition. The nodes were defined as the pixels at the same gray levels and correlations between nodes were based on the absolute difference of two gray levels and gray-level co-occurrence within the 8-connected neighbourhood of a pixel. Pixel distribution, variation of stroke width and size of connected components were extracted as features from the binary sub-images. The text sub-images were obtained by a classifier trained with the above features. The final binarisation result was the combination of all text sub-images. In our approach,

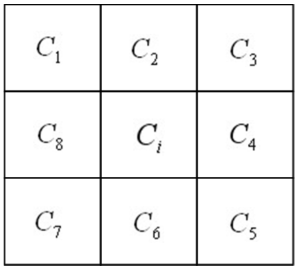


Figure 5.9: 8-connected neighbourhood of a pixel with a value C_i. If a pixel with a value $C_j (j = 1, \cdots, 8)$ appears in the 8-connected neighbourhood of the pixel with a value C_i, then there is a co-occurrence pair between pixel values C_i and C_j.

the colours of a text image are assumed to have two polarities, i.e., text colour and background colour. Therefore, only two sub-images, one for text and the other for background, need to be produced. In our experiment, the selected channel image is mapped into an undirected weighted graph and the definitions of nodes and weights in [75] are adopted to construct the graph.

Definition 1. A node is a group of pixels that have the same pixel value C_i in the selected channel image.

Definition 2. The weight of the link between two nodes at pixel values C_i and C_j, denoted by W_{C_i,C_j}, is defined by

$$W_{C_i,C_j} = \begin{cases} 0 & if |C_i - C_j| > 16, i \neq j, \\ exp(-\frac{|C_i - C_j|}{16}) \times \sqrt{\frac{2N_{C_i,C_j}}{H_{c_i} + H_{c_j}}} & otherwise, \end{cases} \tag{5.1.10}$$

where H_{C_i} and H_{C_j} are the numbers of pixels of the selected channel image at pixel values C_i and C_j respectively, and N_{C_i,C_j} represents the number of 8-connected neighborhood co-occurrence pairs between pixel values C_i and C_j in the selected channel image. See Figure 5.9 for the definition of a co-occurrence pair in a neighborhood.

Definition 2 above implies that a higher weight reflects a stronger correlation between two nodes that have pixels with similar values and closer spatial locations. By Definition

2, two nodes having large enough pixel value difference are deemed to have no correlation as we assume that foreground (or background) pixels should have similar values. The value difference 16 is chosen empirically. In Definition 2, the weight W_{C_i,C_j} increases when the value of $|C_i - C_j|$ decreases because nodes with similar values should have strong correlation. Meanwhile, the pixels constituting a foreground (e.g. text strokes) locate adjacently to each other. A text pixel normally has other text pixels in its local neighbourhood. Thus, the number of co-occurrence pairs N_{C_i,C_j} reflects closeness of nodes spatially. Moreover, the weight is defined to be inversely proportional to H_{C_i} and H_{C_j} so that two nodes will have a weak correlation when they have a larger number of pixels. For scene text images, due to the relative uniformity nature of foreground and background, the numbers of pixels having similar pixel values are large, i.e., the H_{c_i} corresponding to the dominant foreground values and the H_{c_j} corresponding to the dominant background values are large.

Cutting Graph

The selected channel image is mapped to the constructed graph, and then the graph is cut into two parts in order to binarise a text image.

The cost of partition is evaluated by the criteria in normalised cut [99]. The cost of partitioning a graph G into two disjoint sets A and B ($A \bigcup B$) is expressed as:

$$Ncut(A,B) = \frac{\sum\limits_{C_i \in A, C_j \in B} W_{C_i,C_j}}{\sum\limits_{C_i \in A, C_j \in G} W_{C_i,C_j}} + \frac{\sum\limits_{C_i \in A, C_j \in B} W_{C_i,C_j}}{\sum\limits_{C_i \in B, C_j \in G} W_{C_i,C_j}}, \tag{5.1.11}$$

where $G = A \cup B$ is the constructed graph of an image. The cost to cut the graph, denoted by $S(C_x)$, is computed by finding the optimal position C_{min} such that

$$C_{min} = \arg\min_{C_x} S(C_x). \tag{5.1.12}$$

In order to cut the graph and hence to separate the selected channel image into text region A and background region B, the cost function $S(C_x)$ is defined as

$$S(C_x) = \begin{cases} 0 & H_{C_x} > 0, \sum\limits_{C_i \in A, C_j \in B} W_{C_i,C_j} = 0 \\ Ncut(A,B) & elsewise. \end{cases} \tag{5.1.13}$$

where W_{C_i,C_j} is the weight of the link connecting node i and node j, as defined in Equation 5.1.10, H_{C_x} is the number of pixels having pixel value C_x, A and B are defined by $A = \{C_1, \cdots, C_x\}, B = \{C_{x+1}, \cdots, C_n\}, C_x \in \{C_1, \cdots, C_{n-1}\}$, n is the number of nodes in the constructed graph.

5.2 Colour-Based Text Binarisation

In Section 5.1, a thresholding-based text binarisation algorithm was proposed by using colour channel selection and normalised cut. It is a monochrome image binarisation method which relies on discontinuity or homogeneity of pixel values of colour components in the selected colour channel image that contributes most for binarisation. Compared with those methods based on gray-level information, this method makes a use of the information of the colour channel that contributes most for binarisation. However, it ignores the contributions from the other two colour channels. In order to make a full use of colour information which is more informative than monochrome, a simple and effective text binarisation method based on colour information is presented in this section. A new connected component-based text validation measure is proposed for finding the most possible cluster of text after K-means clustering ($K = 3$). In order to choose the better binarisation result obtained from the Euclidean Distance-based and the Cosine Similarity-based 3-means clustering, an objective segmentation evaluation method that describes both the intra-region homogeneity and the inter-region contrast is also proposed. The procedure of our colour-based text binarisation is shown in Figure 5.10.

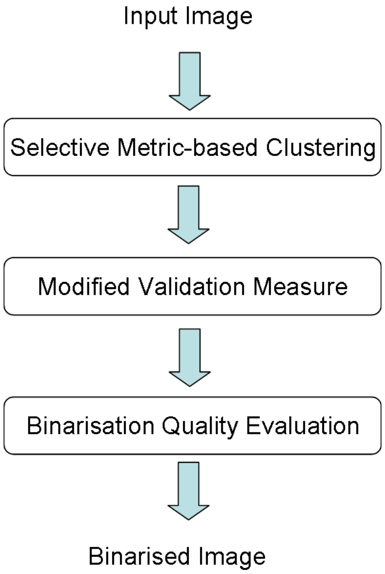

Input Image
Selective Metric-based Clustering
Modified Validation Measure
Binarisation Quality Evaluation
Binarised Image

Figure 5.10: The flow chart of our colour-based text binarisation method.

5.2.1 Selective Metric-based Clustering

According to the extensive comparison of various colour spaces in [69], RGB colour space has demonstrated better performance than most of other colour spaces in classifying pixels when the Euclidean distance D_{eucl} and a cosine-based similarity S_{cos} are complementary clustering distances. The Euclidean distance can deal with text images with no degradations, while a cosine-based similarity can handle text images with degradations. Following [9], the 3-means clustering algorithm is applied to a text image with Euclidean Distance D_{eucl} and Cosine Similarity S_{cos} respectively. The pixels of a text image are classified into textual foreground cluster, background cluster and noise cluster by a 3-means clustering algorithm. The Euclidean Distance D_{eucl} and the Cosine Similarity S_{cos} are defined as:

$$D_{eucl}(\vec{x}_1, \vec{x}_2) = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}, \tag{5.2.1}$$

and

$$S_{cos}(\vec{x}_1, \vec{x}_2) = 1 - \Big(\frac{\vec{x}_1 \cdot \vec{x}_2}{\|\vec{x}_1\| \cdot \|\vec{x}_2\|}\Big)\Big(1 - \frac{\|\vec{x}_1\| - \|\vec{x}_2\|}{max\big(\|\vec{x}_1\|, \|\vec{x}_2\|\big)}\Big). \tag{5.2.2}$$

In Equation 5.2.1 and Equation 5.2.2, $\vec{x}_1 = (R_1, G_1, B_1)^T$ and $\vec{x}_2 = (R_2, G_2, B_2)^T$ are colour vectors in RGB space. R_i, G_i and B_i ($i = 1, 2$) are the red, green and blue colour components respectively.

It is observed that the size of each of the character regions in foreground is usually more regular than those in background and noise regions. A text validation measure M has been proposed in [9] in order to find the most possible foreground cluster, which is defined as:

$$M = \sum_{i=1}^{N} \Big\| area_i - \frac{1}{N}\big(\sum_{i=1}^{N} area_i\big)\Big\|, \tag{5.2.3}$$

where N is the number of CCs and $area_i$ refers to the area of the connected component i. The cluster that has the highest pixel occurrence on the image border is chosen as the background. The M values of the remaining two clusters are denoted as M_1 and M_2 respectively. Meanwhile, the M value for the merged cluster of these two clusters is also computed as M_3. The cluster with the smallest value among M_1, M_2 and M_3 is selected as the text cluster. Between the two binarisation results obtained by 3-means clustering with two distance metrics, the better result is chosen by a step of character segmentation-by-recognition using Log-Gabor filters.

Modification of Validation Measure

The text measurement M defined in Equation (5.2.3) provides a method to evaluate the uniformity of the sizes of different connected components. However, since it simply sums up the terms each representing the absolute difference between a CC's size and the mean size of all CCs, it does not accurately reflect the uniformity of different sets of CCs' sizes, especially when their sizes are significantly different. Hence, such a definition often makes wrong decisions for the selection of text cluster because data in different scales are not comparable. This can be seen in the example shown in Figure 5.11.

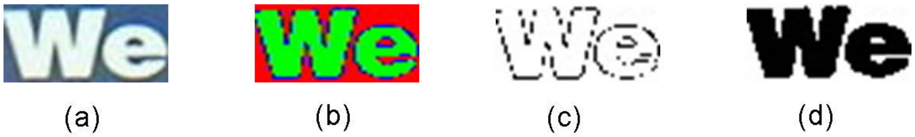


Figure 5.11: A comparison between the binarisation results obtained using M metric in [9] and that using the proposed M_{norm} metric with Euclidean Distance. (a) Original image. (b) 3-means clustering result with Euclidean Distance. Here, the green, red and blue colours represent the textual foreground cluster, background cluster and the noise cluster respectively. (c) Binarised text (in black) by using M. (d) Binarised text (in black) by using M_{norm}. (Best viewed in colour).

In Figure 5.11(b), the red cluster is the background cluster, the green cluster is the textual foreground cluster and the blue cluster lying around the boundary of the textual foreground cluster is the noise cluster. The background cluster (in red) is selected as the cluster has the highest occurrence rate on the border of a text image. There are many small CCs in the noise cluster, two big CCs in the textual foreground cluster (in green) and two big CCs in the cluster merging the noise cluster (in blue) and textual foreground cluster (in green). Following the validation measure M and the rules of selecting the text cluster in [9], the noise cluster is judged as the text cluster. Due to the fact that the sizes of small CCs and the sizes of big CCs are not comparable, the sum of the differences as described in Equation 5.2.3 for the cluster having small CCs is less than that in the cluster having big CCs. Therefore, the cluster with small CCs is chosen as the text cluster like the case illustrated in Figure 1. After performing 3-means clustering with Euclidean Distance in Equation 5.2.1 and Cosine Similarity in Equation 5.2.2, we modify the definition of the CCA-based validation measure for finding the text cluster and define a new measurement denoted as M_{norm} in Equation 5.2.4.

$$M_{norm} = \sum_{i=1}^{N} \left\| \frac{area_i}{A} - \frac{1}{N}(\sum_{i=1}^{N} \frac{area_i}{A}) \right\|, \tag{5.2.4}$$

where A represents the total area of all CCs in a cluster, N is the number of CCs and $area_i$ refers to the area of the CC i. As defined in Equation (5.2.4), by normalising the areas of each CC, the sizes of the CCs in the clusters with significantly different sizes become in a same scale. Similar to the selection rules in [9], the cluster with the smallest value of M_{norm} is determined as the text cluster.

Binarisation Quality Evaluation

After a text image is firstly processed by running 3-means clustering algorithm with two different metrics in Equation 5.2.1 and Equation 5.2.2. Using the proposed measurement M_{norm}, two binarisation results are obtained. Next, objective binarisation evaluation needs to be performed in order to judge the quality of different binarisation results which is feasible in real-time applications. For this purpose, we propose an objective binarisation evaluation method which simultaneously considers intra-region uniformity and inter-region disparity in order to choose the final binarisation result. The intra-region uniformity refers to the homogeneous property within text or background region, while the inter-region disparity refers to the difference along the border between text and background region. Inspired by [131], we define below a metrics E using the local information of pixels to evaluate the two binarisation results according to the two metrics in Equation 5.2.1 and Equation 5.2.2 respectively.

$$E = \frac{C}{UT + UB}, \tag{5.2.5}$$

where C represents the contrast at the border between text and background, UT represents the uniformity of text and UB represents the uniformity of background. C, UT and UB are defined in Equations (5.2.6), (5.2.7) and (5.2.8) respectively.

$$C = \frac{\sum\limits_{(i,j)\in C_{bd}} \big(max\big(D^R_{(i,j)}\big) + max\big(D^G_{(i,j)}\big) + max\big(D^B_{(i,j)}\big)\big)}{3 \cdot 255 \cdot N_{bd}}. \tag{5.2.6}$$

$$UT = \frac{\sum\limits_{(i,j)\in C_{t}} \big(max\big(D^R_{(i,j)}\big) + max\big(D^G_{(i,j)}\big) + max\big(D^B_{(i,j)}\big)\big)}{3 \cdot 255 \cdot N_{t}}. \tag{5.2.7}$$

$$UB = \frac{\sum\limits_{(i,j)\in C_{bk}} \big(max\big(D^R_{(i,j)}\big) + max\big(D^G_{(i,j)}\big) + max\big(D^B_{(i,j)}\big)\big)}{3 \cdot 255 \cdot N_{bk}}. \tag{5.2.8}$$

In Equation 5.2.6, C_{bd} is the set of pixels belonging to the border of text and background, and N_{bd} is the number of pixels in C_{bd}. In Equation 5.2.7, C_t is the set of pixels belonging to text, and N_t is the number of pixels in C_t . In Equation 5.2.8, C_{bk} is the set of pixels belonging to background, and N_{bk} is the number of pixels in C_{bk}. In Equation 5.2.6, Equation 5.2.7 and Equation 5.2.8, $\big(max\big(D^R_{(i,j)}\big)$, $\big(max\big(D^G_{(i,j)}\big)$ and $\big(max\big(D^B_{(i,j)}\big)$ refer to the maxima of the differences between the value of pixel at (i, j) and those of its 4-neighbours in R, G and B channels respectively.

Between the two text binarisation results corresponding to the two metrics, the one with the greater E value is chosen as the final binarisation result.

5.3 Experimental Results of the Proposed Methods

The performance of the proposed gray level-based and colour-based methods are evaluated qualitatively and quantitatively. The qualitative evaluation is undertaken by comparing binarisation results of different binarisation algorithms intuitively. The Otsu's method [78], an edge-based binarisation method [10], the proposed gray level-based method and the proposed colour-based method are implemented for comparison. Text images from the ICDAR2003 Robust Word Recognition dataset are used for testing. Some binarisation examples for qualitative evaluation are shown below (from Figure 5.12 to Figure 5.16).

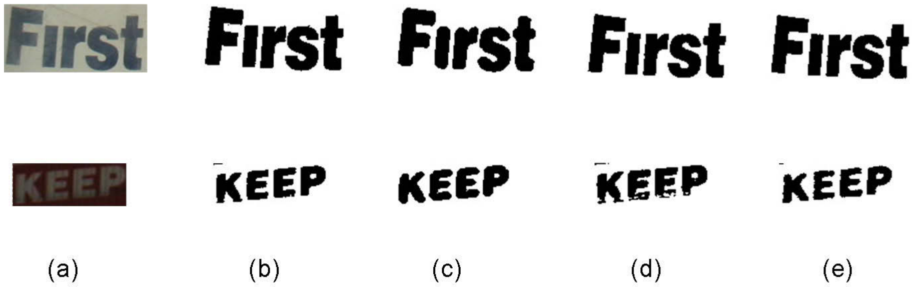


Figure 5.12: Simple cases. (a) Original image. (b) Otsu's method. (c) The method in [10]. (d) The proposed gray level-based method. (e) The proposed colour-based method.

The comparisons are under various circumstances including uneven lighting (e.g., Figure 5.13), complex background (e.g., Figure 5.14) and highlight (e.g., Figure 5.15). In each example from Figure 5.12 to Figure 5.16, the first column is the original image, the second column is the binarisation result using Otsu's method, the third column is the binarisation result using the edge-based binarisation method [10], the fourth column is the binarisation result using the proposed gray level-based method and the fifth column is the binarisation result using the proposed colour-based method.

As shown in the examples, for the simple cases as shown in Figure 5.12, the proposed

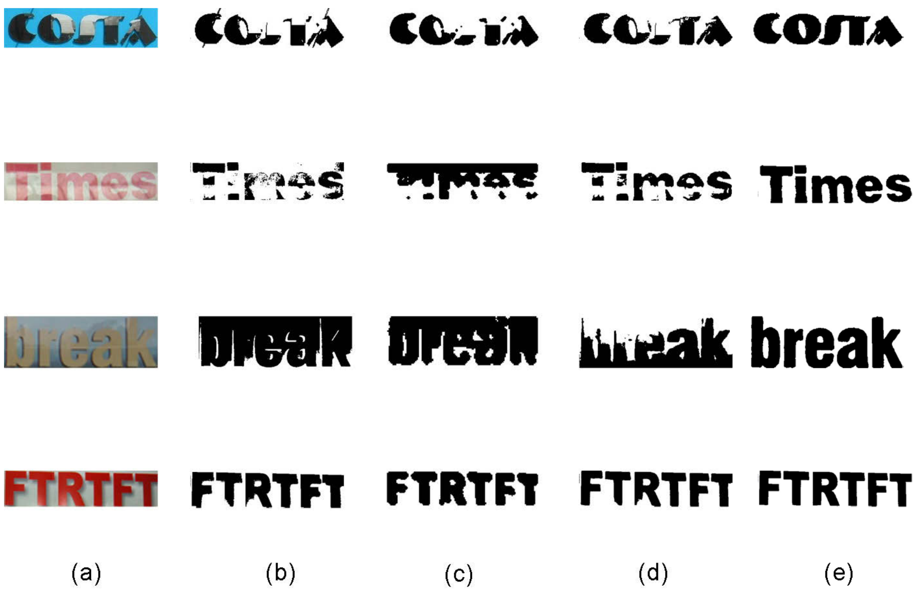


Figure 5.13: Uneven lighting cases. (a) Original image. (b) Otsu's method. (c) The method in [10]. (d) The proposed gray level-based method. (e) The proposed colour-based method.

methods get as good binarisation results as the other two methods. The proposed colour-based method achieves the best binarisation result in text images with uneven lighting. The use of colour information can describe the feature of colour text image and make the binarisation less sensitive to uneven lighting. This keeps the integrity of text characters as shown in Figure 5.13. The proposed validation measure of our colour-based method provides robust binarisation from complex background and shadow as shown in Figure 5.14. In Figure 5.15, the text images are degraded by highlight. It can be seen that our colour-based method generates the best binarisation results visually. In Figure 5.16, both the Otsu's method and the proposed methods can acquire a good binarisation of the text with low contrast, while the edge-based method misses four letters (third row) due to the fact that there is no seed generated within the edges of the missing letters. Overall, the proposed gray level-based method is competitive with the Otsu's method [78] and the edge-based binarisation method [10]. Our colour-based method can get the best results among

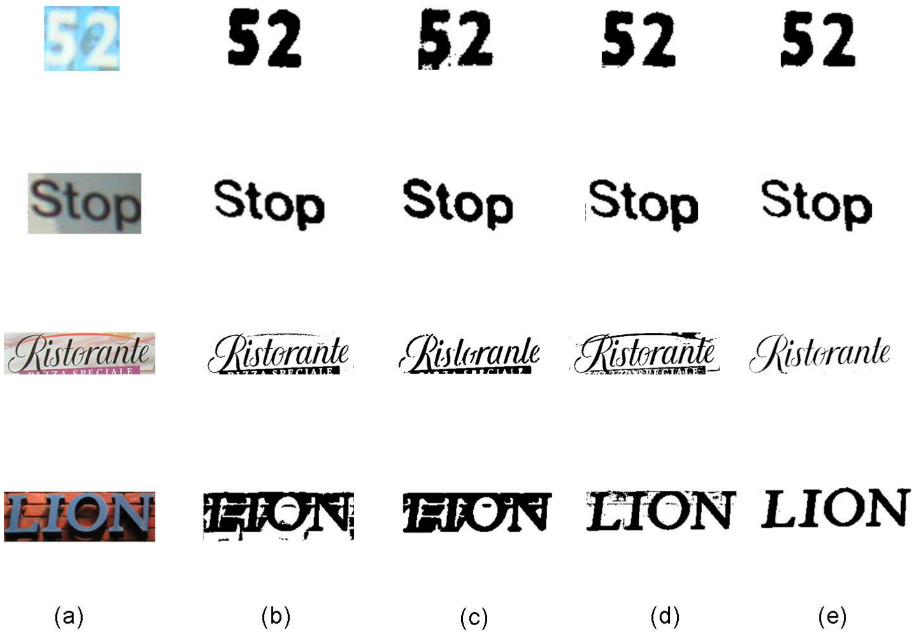


Figure 5.14: Complex background cases. (a) Original image. (b) Otsu's method. (c) The method in [10]. (d) The proposed gray level-based method. (e) The proposed colour-based method.

the algorithms under comparison.

For further evaluate the performance of the proposed algorithms in a wider range of text images, we compare the character recognition rates of different algorithms. The binarised text images are fed to OCR softwares for recognition. We use sample images from the ICDAR2003 Robust Word Recognition dataset as in the experiment in [40]. There are 171 natural scene text images in total. These images also have the same kinds of degradations as in the qualitative evaluation. Our methods are compared with Otsu's method [78], Sauvola's method [132], Niblack's method [133], Kittler's method [134], Thillou's method [135] and Mishra's method [40]. The binarisation results are sent to commercial OCR ABBYY fine reader for recognition. The comparison results are listed in Table 5.1.

The comparison in Table 5.1 shows that the proposed gray level-based method has

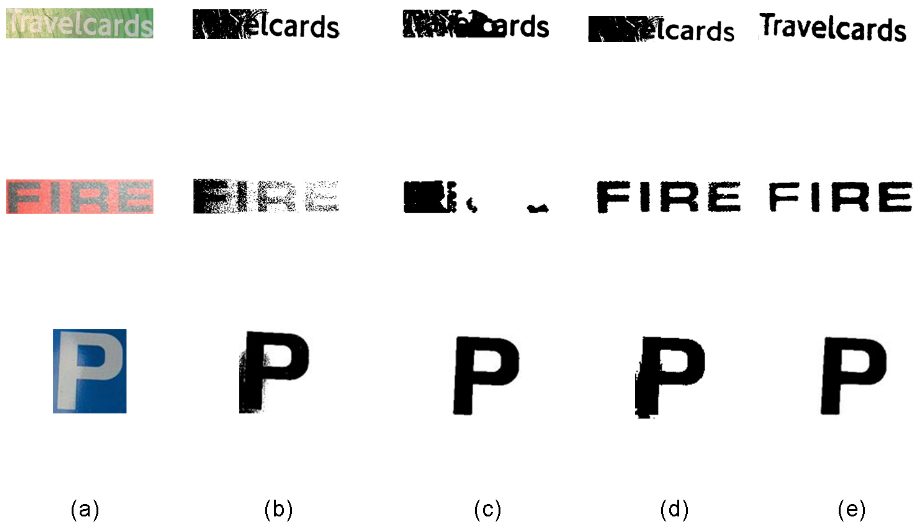


Figure 5.15: Highlight cases. (a) Original image. (b) Otsu's method. (c) The method in [10]. (d) The proposed gray level-based method. (e) The proposed colour-based method.

Table 5.1: Comparison of character recognition rates

Methods	Character Recognition Rate
Our colour-based method	**63.37%**
Mishra et al. [40]	60.14%
Our gray level-based method	**57.14%**
Otsu [78]	51.98%
Otsu + CT [135]	51.74%
Sauvola [132]	51.63%
Niblack [133]	42.31%
Kittler et al. [134]	49.88%

higher character recognition rate than the well-known thresholding-based methods. This reason is that the selected colour channel image has more distinctive pixel value difference between foreground and background than the gray level map directly converted from colour information. Our colour-based method achieves the highest character recognition rate due to its capability of generating clear binarised text images under various contamination situations.

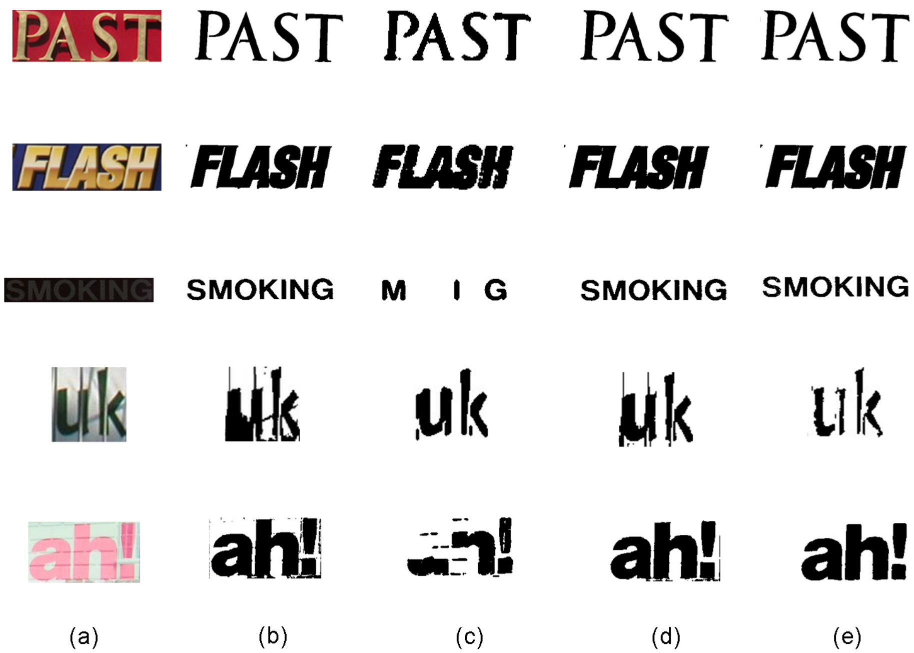


Figure 5.16: Other cases. (a) Original image. (b) Otsu's method. (c) The method in [10]. (d) The proposed gray level-based method. (e) The proposed colour-based method.

5.4 Summary

This chapter proposes one gray level-based text binarisation method and one colour-based text binarisation method. Section 5.1 presents a text binarisation method using graph model construction based on a selected channel image. To circumvent the scenario that different colours may be converted into an identical gray-level when a colour text image is mapped into its gray-level map, we have used the colour channel image which contributes most to binarisation. The selected channel image is the one having the biggest value difference between the two pixels corresponding to two main peaks in the histogram. The definitions of the “node” and the “weight” of the graph model are also given in Section 5.1.4. Different from the traditional gray-level based methods, the proposed method considers both range-domain and spatial-domain for graph construction, and the experimental results show the effectiveness of the proposed method.

For a full use of colour information, a colour-based image binarisation method is presented in Section 5.2. This algorithm consists of three steps: 3-means clustering with two distance metrics, modified CCA-based validation measure and binarisation evaluation method. R, G and B colour information are used as the feature vector for every pixel to cluster the pixels of the text image into three clusters using the Euclidean distance and a cosine-based similarity respectively. A new validation measure is proposed based on normalised size of connected components to assess the possibility of being the text cluster. Finally, the better one of the two candidate binarisation results is selected using the proposed binarisation evaluation method when taking into account the intra-region uniformity and the inter-region disparity. Exemplar results demonstrate that the proposed algorithm get satisfactory text binariation results even when the text images are under various kinds of degradations.

Chapter 6

Conclusions and Future Work

In the first chapter, the general knowledge, applications and existing problem of text information extraction systems have been introduced. The second chapter has presented the relevant research work on text detection and text segmentation. From Chapter 3 to Chapter 5, all of our research work with the technical details have been illustrated through discussion, processing procedures and experimental results comparison. In this chapter, the conclusions will be drawn in Section 6.1 and the future work under incubation will be discussed in Section 6.2.

6.1 Conclusions

This thesis focuses on the investigation and development of text detection and text binarisation algorithms for text in born-digital images and natural scene images. Our methods for text detection and text binarisation have shown the superiority to other methods through the experimental results using benchmark text detection and text binarisation datasets.

6.1.1 Text Detection

The first text detection algorithm focuses on detecting text in born-digital images. Characters in text lines align spatially close to each other and the transition between text and background is quite dense in text regions. The text regions can be effectively highlighted by Maximal Gradient Difference (MGD) values calculated from the whole gray-level image. The multiple layer image scheme favours the algorithm to detect text lines with both strong and weak contrast. With discriminative capability of the proposed texture-based feature descriptor, the detected non-text regions in the coarse detection step can be effectively removed. The experimental results show that our algorithm outperforms other algorithms in a competition using a benchmark born-digital image dataset.

An approach to detect text lines from images captured in natural scene environment is also proposed. Natural scene text typically have salient contrast against the background and the strokes of a character have uniform colour. These characteristics make it successful to generate a connected component for each character by the implementation of maximally stable extremal regions (MSER). A classifier trained by geometry-based, stroke-based, gradient-based and colour-based features is applied for text/non-text MSER classification. As the false positive of the MSER classification is quite low, it becomes reliable to bring back the misclassified single character MSERs by the correctly classified single character MSERs. The character MSERs in small text lines tend to overlap each other. Hence, the MSERs which include multiple touching characters are treated as text line connected components and the corresponding text lines are retrieved by a text line classifier. The combination of these two text MSER retrieval strategies improve the recall rate of our algorithm. A false alarm elimination step is performed for enhance the detection precision. The final detection results and the comparisons with other natural scene text detection methods have illustrated the good performance of the proposed algorithm.

6.1.2 Text Binarisation

A gray level-based method and a colour-based method are proposed on the research on text binarisation. The gray level-based method automatically choose the colour channel image that has the most distinctive difference between foreground pixel values and background pixel values. Both pixel values and the spatial locations of the pixels belonging to the same class (i.e. foreground and background) are considered in the definition of weight in graph construction. The colour-based method uses two clustering distances for clustering the pixels of text images into three groups. A new validation measure is proposed to assign each of these three groups of pixels to be foreground or background. As there are two binarisation results for a text image, a binarisation quality evaluation is performed for selecting the better binarisation result in terms of intra-region uniformity and inter-region disparity. The experimental results show that the proposed methods can achieve better binarisation results than classical binarisation methods. Both qualitative and quantitative evaluations show the effectiveness of the proposed methods, especially the colour-based method which outperforms other methods under comparison.

6.2 Future Work

Aiming at developing algorithms with higher accuracy and efficiency, more efforts will be made in our future work in the following aspects: refinement of pattern recognition scheme (Section 6.2.1), parallel computing for reducing running time (Section 6.2.2) and text recognition (Section 6.2.3).

6.2.1 Refinement of Pattern Recognition Scheme

Pattern recognition consists of image preprocessing, feature generation, feature selection (or extraction) and machine learning. Image preprocessing is to make the interesting information of the objects in digital image stand out and suppress the redundant information. Feature generation uses delicately designed discriminate feature vector(s) to depict the characteristics of the objects or the behaviour patterns that we are interested in. Feature selection and feature extraction, two general approaches for feature space dimensionality reduction, choose a better feature space based on the original feature space. Features with less dimensionalities and more representative power favour a pattern recognition framework in the aspects of effectiveness and efficiency. Machine learning produces classifiers trained by training data to discriminate from the interests and non-interests. The specific approach of machine learning method influences the training time and generalization capability of the classifiers. Hence, the pursuit of better machine learning methods is necessary.

Feature Selection/Extraction

A lot of different types of features have been proposed for various computer vision and image processing applications. Single type of feature may not be enough to depict the characteristics of the interests. Therefore, multiple types of features are used together. However, more features do not mean better description of the interests. The curse of dimensionality [136] states that there is a maximum number of features above which the performance of a classifier will degrade rather than improve. In spite of this, the lost information due to discarding some features can be compensated by a more accurate mapping in lower-dimensional space in most cases. It comes to finding the most expressive features as well as reducing the dimensionality of feature vectors and this can be achieved by feature selection/extraction. The difference between feature selection and extraction is: feature extraction techniques transform the existing features into a lower dimensional space and feature selection techniques select a subset of the existing features without a transformation. In other words, feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection is a more general method, but there are some occasions where feature selection is necessary:

- Features that may be expensive to obtain.
- Extracting meaningful rules from the classifier.
- Having many features and relatively few samples.

In our future work, feature extraction/selection will be applied for choosing those more "better" features to express the object of interest and reduce the time consumed in the classifier training phrase.

Random Forest Training

Supervised learning algorithms search through a hypothesis space to find a suitable hypothesis that has good prediction ability with a particular problem. Ensembles combine multiple hypotheses to form a better hypothesis. This means that a strong classifier can be generated from many weak classifiers. Random forest [137] is an ensemble classifier which is composed of many decision trees and outputs the class that is the mode of the classes output by individual trees. As stated in [138], random forest performs very well compared to many other classifiers including discriminant analysis, support vector machines and neural networks. In addition, random forest is of the following features and advantages [139]:

- One of the most accurate learning algorithms.

- Running efficiently on large databases.
- Being able to process thousands of input variables without variable deletion.
- Being able to estimate the importance of different variables
- Generation of an internal unbiased estimate of the generalization error.
- Having an effective method for estimating missing data and maintains accuracy.
- Being able to balance error in class population unbalanced data sets.
- Providing information regarding the relationship between the variables and classification.
- Being able to be extended to unsupervised clustering.

Since random forest has such many good features, it will be helpful to machine learning-based classification task in the future research work.

6.2.2 Parallel Computing for Reducing Running Time

As the quick development of computer technology, tremendous amount of data are generated, transmitted and processed. Consequently, more and more data will exponentially increase which leads to extremely quick growth of computation running time. Even a single very high-performance computer cannot deal with a very large amount of data with its limited resource. This has become a significant problem in real-time applications. High processing speed is always an inevitable requirement to any algorithms for real-time application. Parallel computing [140] is a form of computation in which many calculations are run at the same time. Basically, parallel computing simultaneously uses multiple computation resources to solve a computational task through:

- Using multiple CPUs.
- Dividing a problem into separate portions that can be solved at the same time.
- Breaking each portion down to a series of instructions.
- Runing instructions from each portion concurrently on different CPUs.

Parallel computing had been discussed on its application on digital image processing [141–143]. Point operators, local operators, smoothing, edge detection, edge thinning, corner detection, Hough Transform, morphological operators, Fast Fourier Transform, motion detection and image segmentation, which are common processes in computer vision and image processing algorithms, are suitable for concurrent parallel implementations [142]. In the viewpoint of machine learning, parallel computing can be used for training a classifier. For example, when various types of features are used to train several support vector machine classifiers and the final decision is fused by the results of these classifiers, training these classifiers simultaneously can save time.

Since text information extraction system development is real-time application oriented, our research work will be benefited from the time-saving property of parallel computing.

6.2.3 Text Recognition

In this thesis, only the first two components of the text information extraction system have been considered. When there is the demand of “reading” texts in a practical application, the texts in text lines should be converted from image format to digital format. Hence, the text recognition component is necessary to be included for an integrated system. Text recognition is composed of two phrases: character isolation and character recognition. Character isolation, in a concise form, means separating all of the letters, characters or numbers into individual ones. Currently, most algorithms designed for text recognition binarise the text line images (black for text and white for background), then feed the binarised text line images into an optical character recognition software for recognition. Nevertheless, noise, uneven lighting and reflection may degrade the quality of the binarised image and, further, affect the recognition rate. It has been shown in [144] that dividing the characters in a text line into individual characters can improve the recognition accuracy. The colour and contrast of a single character have more consistency than that of all characters in a text line image. Taking this situation into account, character isolation needs to be embedded in the whole system.

Optical character recognition (OCR) [145] is a very mature technique for recognising printed text in document. Printed text usually has normal fonts, clear background, less degradation from highlight or uneven lighting. All these characteristics make the high recognition rate. However, it is another story for natural scene text recognition. Natural scene text may have a specially designed style of the font. The reflection caused by light makes texts do not have consistent appearances. The camera position may be slanted when a text image is taken. These cases bring more difficulties in the recognition task. Therefore, further research on text recognition is necessary to carry out.

Author's Publication list

1. Chao Zeng, Wenjing Jia and Xiangjian He, 'A novel framework of detecting text from natural scene images', Pattern Recognition. Submitted.

2. Chao Zeng, Wenjing Jia, Xiangjian He and Liming Zhang, 'Text Detection in Born-Digital Images Using IT-LBP', Journal of Algorithms & Computational Technology (JACT), 2013. Accepted on 23rd August 2013.

3. Chao Zeng, Wenjing Jia and Xiangjian He, 'Text Detection in Born-Digital Images Using Multiple Layer Images', IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Canada, May, 2013.

4. Chao Zeng, Wenjing Jia and Xiangjian He, 'An Algorithm for Colour-Based Natural Scene Text Segmentation', ICDAR Workshop on Camera-Based Document Analysis and Recognition (CBDAR), Lecture Notes in Computer Science, Volume 7139, pp.58-68, 2011.

5. Chao Zeng, Wenjing Jia, Xiangjian He and Jie yang, 'Graph-based Text Segmentation Using A Selected Channel Image', International Conference on Digital Image Computing: Techniques and Applications (DICTA), Sydney, pp.535-539, December, 2010.

6. Chao Zeng, Wenjing Jia, Xiangjian He and Min Xu, 'Recent Advances on Graph-Based Image Segmentation Techniques', in Book *Graph-Based Methods in Computer Vision: Developments and Applications*, *IGI Global*, pp. 140-154, July, 2012.

7. Jia Wenjing, Chao Zeng, Yongxia Ao, Xiangjian He and Qiang Wu, 'Automatic Detection and Extraction of Sign Text from Outdoor Scene: A Contemporary Review', Journal of Yunnan University Nationalities (Natural Sciences Edition), Volume 19, No. 3, pp. 157-161, 2010.

8. Chao Zeng, 'Graph-Based Scene Text Segmentation Using Colour Information', UTS FEIT Research Showcase, Sydney, June, 2010.

Bibliography

[1] D. Karatzas, S. Mestre, J. Mas, F. Nourbakhsh, and P. Roy, “Icdar 2011 robust reading competition challenge 1: Reading text in born-digital images (web and email),” in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1485 – 1490, 2011.

[2] A. Shahab, F. Shafait, and A. Dengel, “Robust competition challenge 2: Reading text in scene images,” in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1491–1496, 2011.

[3] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[4] Y. Zeng, D. Samaras, W. Chen, and Q. Peng, “Topology cuts: a novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images,” *Computer Vision and Image Understanding*, vol. 112, no. 1, pp. 81–90, 2008.

[5] L. Grady, “Random walks for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[6] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[7] Y. Wei and C. Lin, “A robust video text detection approach using svm,” *Expert Systems with Applications*, vol. 39, no. 12, pp. 10832–10840, 2012.

[8] H. Zhang, J. Fritts, and S. Goldman, "Image segmentation evaluation: a survey of unsupervised methods," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 260–280, 2008.

[9] C. M. Thillou and B. Gosselin, "Color text extraction with selective metric-based clustering," *Computer Vision and Image Understanding*, vol. 107, no. 1-2, pp. 97–107, 2007.

[10] Z. Zhou, L. Li, and C. L. Tan, "Edge based binarization for video text images," in *International Conference on Pattern Recognition (ICPR)*, pp. 133–136, 2010.

[11] "License plate recognition. `http://www.platerecognition.info/1106.htm`,"

[12] M. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: state of the art and challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications*, pp. 1–19, 2006.

[13] M. Xu, J. Wang, M. Hasan, X. He, C. Xu, H. Lu, and J. Jin, "Using context saliency for movie shot classification," in *IEEE International Conference on Image Processing (ICIP)*, pp. 3653–3656, 2011.

[14] D. Wang, Q. Tian, S. Gao, and W. Sung, "News sports video shot classification with sports play field and motion features," in *IEEE International Conference on Image Processing (ICIP)*, vol. 4, pp. 2247–2250, 2004.

[15] S. Wang, S. Jiang, Q. Huang, and W. Gao, "Shot classification for action movies based on motion characteristics," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2508–2511, 2008.

[16] C. Lee, K. Jung, and H. Kim, "Automatic text detection and removal in video sequences," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2607–2623, 2003.

[17] H. Shiratori, H. Goto, and H. Kobayashi, "An efficient text capture method for moving robots using dct feature and text tracking," in *International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 1050–1053, 2006.

[18] M. Tanaka and H. Goto, "Autonomous text capturing robot using improved dct feature and text tracking," in *International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2, pp. 1178–1182, 2007.

[19] C. Yi and Y. Tian, "Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification," *IEEE Transactions on Image Processing*, vol. 2, no. 9, pp. 4256–4268, 2012.

[20] E. Tekin, J. Coughlan, and H. Shen, "Real-time detection and reading of led/lcd displays for visually impaired persons," in *IEEE Workshop on the Applications of Computer Vision (WACV)*, pp. 491–496, 2011.

[21] "Advanced driver assistance systems. `http://en.wikipedia.org/wiki/Advanced_driver_assistance_systems`,"

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

[23] H. Chen, S. Tsai, G.Schroth, D. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2609–2612, 2011.

[24] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3538–3545, 2012.

[25] B. Epshtein, E. Ofek, and Y. Wexker, “Detecting text in natural scenes with stroke width transform,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2963–2970, 2010.

[26] C. Yi and Y. Tian, “Text string detection from natural scenes by structure-based partition and grouping,” *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2594–2605, 2011.

[27] “Blob detection. `http://http://en.wikipedia.org/wiki/Blob_detection`,”

[28] “Maximally stable extremal regions. `http://en.wikipedia.org/wiki/Maximally_stable_extremal_regions`,”

[29] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, “Icdar 2003 robust reading competitions,” in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 682–687, 2003.

[30] S. Lucas, “Icdar 2005 text locating competition results,” in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 80–84, 2005.

[31] P. Shivakumara, T. Phan, and C. Tan, “A laplacian approach to multi-oriented text detection in video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 412–419, 2011.

[32] D. Chen, H. Bourlard, and J. Thiran, “Text identification in complex background using svm,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. II–621 – II–626, 2001.

[33] M. Anthimopoulos, B. Gatos, and I. Pratikakis, “A two-stage scheme for text detection in video images,” *Image and Vision Computing*, vol. 28, no. 9, pp. 1413–1426, 2010.

[34] C. Jung, Q. Liu, and J. Kim, "A stroke filter and its application to text localization," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 114–122, 2009.

[35] P. Shivakumara, T. Phan, and C. Tan, "A gradient difference based technique for video text detection," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 156–160, 2009.

[36] E. Wong and M. Chen, "A new robust algorithm for video text extraction," *Pattern Recognition*, vol. 36, no. 6, pp. 1397–1406, 2003.

[37] C. Liu, C. Wang, and R. Dai, "Text detection in images based on unsupervised classification of edge-based features," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 610–614, 2005.

[38] T. Kasar and A. G. Ramakrishnan, "Cococlust: Contour-based color clustering for robust binarization of colored text," in *International Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*, pp. 11–17, 2009.

[39] J. Lim, J. Park, and G. Medioni, "Text segmentation in color images using tensor voting," *Image and Vision Computing*, vol. 25, no. 5, pp. 671–685, 2007.

[40] A. Mishra, K. Alahari, and C. Jawahar, "An mrf model for binarization of natural scene text," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 11–16, 2011.

[41] C. Jung, Q. Liu, and J. Kim, "Accurate text localization in images based on svm output scores," *Image and Vision Computing*, vol. 27, no. 9, pp. 1295–1301, 2009.

[42] W. Pan, T. Bui, and C. Suen, "Text detection from scene images using sparse representation," in *International Conference on Pattern Recognition (ICPR)*, pp. 1–5, 2008.

[43] H.Shen and J. Coughlan, “Finding text in natural scenes by figure-ground segmentation,” in *International Conference on Pattern Recognition (ICPR)*, vol. 4, pp. 113–118, 2006.

[44] P. Silapachote, J. Weinman, A. Hanson, R. Weiss, and M. Matter, “Automatic sign detection and recognition in natural scenes,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, p. 27, 2005.

[45] C. Zhu, W. Wang, and Q. Ning, “Text detection in images using texture feature from strokes,” *Lecture Notes in Computer Science*, vol. 4261, pp. 295–301, 2006.

[46] J. Kim, S. Park, and S. Kim, “Text locating from natural scene images using image intensities,” in *International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2, pp. 655–659, 2005.

[47] Q. Ye, J. Jiao, J. Huang, and H. Yu, “Text detection and restoration in natural scene images,” *Journal of Visual Communication and Image Representation*, vol. 18, no. 6, pp. 504–513, 2007.

[48] J. Park and S. Park, “Detection of text region and segmentation from natural scene images,” *Lecture Notes in Computer Science*, vol. 3804, pp. 666–671, 2005.

[49] W. Kim and C. Kim, “A new approach for overlay text detection and extraction from complex video scene,” *IEEE Transactions on Image Processing*, vol. 18, no. 2, pp. 401–411, 2009.

[50] Q. Liu, C. Jung, S. Kim, Y. Moon, and J. Kim, “Stroke filter for text localization in video images,” in *IEEE International Conference on Image Processing (ICIP)*, pp. 1473–1476, 2006.

[51] K. Subramanian, P. Natarajan, M. Decerbo, and D. Castanon, “Character-stroke detection for text-localization and extraction,” in *International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 33–37, 2007.

[52] J. Zhang and R. Kasturi, "Character energy and link energy-based text extraction in scene images," in *Asian Conference on Computer Vision (ACCV)*, vol. 2, pp. 308–320, 2010.

[53] X. Liu and W. Wang, "Robustly extracting captions in videos based on stroke-like edges and spatio-temporal analysis," *IEEE Transactions on Multimedia*, vol. 14, no. 2, pp. 482–489, 2012.

[54] S. Hanif, L. Prevost, and P. Negri, "A cascade detector for text detection in natural scene images," in *International Conference on Pattern Recognition (ICPR)*, pp. 1–4, 2008.

[55] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, "Large-scale privacy protection in google street view," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2373–2380, 2009.

[56] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. II–366–II–373, 2004.

[57] X. Chen and A. Yuille, "A time-efficient cascade for real-time object detection: with applications for the visually impaired," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28–28, 2005.

[58] H. Zhang, W. Jia, X. He, and Q. Wu, "Learning-based license plate detection using global and local features," in *International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 1102–1105, 2006.

[59] S. Hanif and L. Prevost, "Text detection and localization in complex scene images using constrained adaboost algorithm," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1–5, 2009.

[60] Y.-F. Pan, X. Hou, and C.-L. Liu, “A robust system to detect and localize texts in natural scene images,” in *IAPR International Workshop on Document Analysis Systems (DAS)*, pp. 35–42, 2008.

[61] T. Saoi, H. Goto, and H. Kobayashi, “Text detection in color scene images based on unsupervised clustering of multi-channel wavelet features,” in *International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2, pp. 690–694, 2005.

[62] H. Goto, “Redefining the dct-based feature for scene text detection,” *International Journal on Document Analysis and Recognition*, vol. 11, pp. 1–8, 2008.

[63] T. Kasar, J. Kumar, and A. G. Ramakrishnan, “Font and background color independent text binarization,” in *International Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*, pp. 3–9, 2007.

[64] J. Yu, L. Huang, and C. Liu, “Double-edge-model based character stroke extraction from complex backgrounds,” in *International Conference on Pattern Recognition (ICPR)*, pp. 1–4, 2008.

[65] X. Ye, M. Cheriet, and C. Y. Suen, “Stroke-model-based character extraction from gray-level document images,” *IEEE Transactions on Image Processing*, vol. 10, no. 8.

[66] X. Li, W. Wang, Q. Huang, W. Gao, and L. Qing, “A hybrid text segmentation approach,” in *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 510–513, 2009.

[67] M. Yokobayashi and T. Wakahara, “Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation,” in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 167–171, 2005.

[68] M. Yokobayashi and T. Wakahara, "Binarization and recognition of degraded characters using a maximum separability axis in color space and gat correlation," in *International Conference on Pattern Recognition (ICPR)*, pp. 885–888, 2006.

[69] C. M. Thillou and B. Gosselin, "Color text extraction from camera-based images: the impact of the choice of the clustering distance," in *International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 312–316, 2005.

[70] Y. Song, K. Kim, Y. Choi, H. Byun, S. Kim, S. Chi, D. Jang, and Y. Chung, "Text region extraction and text segmentation on camera-captured document style images," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 172–176, 2005.

[71] C. Mancas-Thillou and B. Gosselin, "Spatial and color spaces combination for natural scene text extraction," in *IEEE International Conference on Image Processing (ICIP)*, pp. 985–988, 2006.

[72] J. Yao, Y. Gao, L. Ma, and Y. Yang, "Scene text extraction based on hsl," *International Symposium on Computer Science and Computational Technology*, pp. 315–319, 2008.

[73] Q. Liu, C. Jung, and Y. Moon, "Text segmentation based on stroke filter," *Proceedings of the 14th Annual ACM International Conference on Multimedia*, pp. 129–132, 2006.

[74] C. Jung, Q. Liu, and J. Kim, "A new approach for text segmentation using a stroke filters," *Signal Processing*, vol. 88, no. 7, pp. 1907–1916, 2008.

[75] J. Li, Y. Tian, T. Huang, and W. Gao, "Multi-polarity text segmentation using graph theory," in *IEEE International Conference on Image Processing (ICIP)*, pp. 3008–3011, 2008.

[76] M. Mignotte, "Segmentation by fusion of histogram-based k-means clusters in different color spaces," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 780–787, 2008.

[77] S. Rao, H. Mobahi, A. Yang, S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *Asian Conference on Computer Vision (ACCV)*, 2009.

[78] N. Otsu, "A threshold selection method from gray-level histogram," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[79] N. Senthilkumaran and R. Rajesh, "Edge detection techniques for image segmentation-a survey of soft computing approaches," *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, pp. 250–254, 2009.

[80] J. Fan, G. Zeng, M. Body, and M. Hacid, "Seeded region growing: an extensive and comparative study," *Pattern Recognition Letters*, vol. 26, no. 8, pp. 1139–1156, 2010.

[81] J. Zhang, J. Zheng, and J. Cai, "A diffusion approach to seeded image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2125–2132, 2010.

[82] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, pp. 105–112, 2001.

[83] V. Grau, A. Mewes, R. K. M. Alcaniz, and S. Warfield, "Improved watershed transform for medical image segmentation using prior information," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 447–458, 2004.

[84] J. Yuan, E. Bae, and X. Tai, "A study on continuous max-flow and min-cut approaches," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2217–2224, 2010.

[85] B. Peng and O. Veksler, "Parameter selection for graph cut based image segmentation," in *British Machine Vision Conference (BMVC)*, 2008.

[86] S. Candemir and Y. Akgul, "Adaptive regularization parameter for graph cut segmentation," *Lecture Notes in Computer Science*, vol. 6111, pp. 117–126, 2010.

[87] X. Han, C. Xu, and J. Prince, "A topology preserving level set method for geometric deformable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 755–768, 2003.

[88] G. Bertrand, "Simple points, topological numbers and geodesic neighborhoods in cubic grids," *Pattern Recognition Letters*, vol. 15, no. 10, pp. 1003–1011, 1994.

[89] B. Price, B. Morse, and S. Cohen, "Geodesic graph cut for interactive image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3161–3168, 2010.

[90] H. Koo and N. Cho, "Graph cuts using a riemannian metric induced by tensor voting," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 514–520, 2009.

[91] Y. Fu, J. Cheng, Z. Li, and H. Lu, "Saliency cuts: an automatic approach to object segmentation," in *International Conference on Pattern Recognition (ICPR)*, pp. 1–4, 2008.

[92] C. Jung, B. Kim, and C. Kim, "Automatic segmentation of salient objects using iterative reversible graph cut," in *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 590–595, 2010.

[93] L. Grady and G. Funka-Lea, "Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials," in *European Conference on Computer Vision Workshop (ECCVW)*, pp. 230–245, 2004.

[94] A. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1–8, 2007.

[95] L. Grady and A. Sinop, "Fast approximate random walker segmentation using eigenvector precomputation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[96] L. Grady, "Multilabel random walker image segmentation using prior models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 763–770, 2005.

[97] S. Andrews, G. Hamarneh, and A. Saad, "Fast random walker with priors using precomputation for interactive medical image segmentation," *Lecture Notes in Computer Science*, vol. 6363, pp. 6–16, 2010.

[98] W. Yang, J. Cai, J. Zheng, and J. Luo, "User-friendly interactive image segmentation through unified combinatorial user inputs," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2470–2479, 2010.

[99] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[100] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15.

[101] L. Xu, W. Li, and D. Schuurmans, “Fast normalized cut with linear constraints,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2866–2873, 2009.

[102] C. Cigla and A. Alatan, “Efficient graph-based image segmentation via speeded-up turbo pixels,” in *IEEE International Conference on Image Processing (ICIP)*, pp. 3013–3016, 2010.

[103] D. Hochbaum, “Polynomial time algorithms for ratio regions and a variant of normalized cut,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 889–898, 2010.

[104] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[105] T. Phan, P. Shivakumara, and C. Tan, “A laplacian method for video text detection,” in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 66–70, 2009.

[106] M. Felhi, N. Bonnier, and S. Tabbone, “A robust skew detection method based on maximum gradient difference and r-signature,” in *IEEE International Conference on Image Processing (ICIP)*, pp. 2617–2620, 2011.

[107] T. Ojala, M. Pietikainen, and D. Harwood, “A comprehensive study of texture measures with classification based on feature distributions,” *Pattern Recognition*, vol. 29, no. 11, pp. 51–59, 1996.

[108] B. Jun and D. Kim, “Robust face detection using local gradient patterns and evidence accumulation,” *Pattern Recognition*, vol. 45, no. 9, pp. 3304–3316, 2012.

[109] B. Zhang, Y. Gao, S. Zhao, and J. Liu, “Local derivative pattern versus local binary pattern face recognition with high-order local pattern descriptor,” *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 533–544, 2010.

[110] S. Moore and R. Bowden, "Local binary patterns for multi-view facial expression recognition," *Computer Vision and Image Understanding*, vol. 115, no. 4, pp. 541–558, 2011.

[111] X. Wang, T. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 32–39, 2009.

[112] L. Nanni, S. Brahnam, and A. Lumini, "Local ternary patterns from three orthogonal planes for human action classification," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5125–5128, 2011.

[113] T. Jabid, M. Kabir, and O. Chae, "Gender classification using local directional pattern (ldp)," in *International Conference on Pattern Recognition (ICPR)*, pp. 2162–2165, 2010.

[114] L. Nanni, A. Lumini, and S. Brahnam, "Survey on lbp based texture descriptors for image classification," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3634–3641, 2012.

[115] M. Anthimopoulos, B. Gatos, and I. Pratikakis, "A hybrid system for text detection in video frames," in *IAPR International Workshop on Document Analysis Systems (DAS)*, pp. 286–292, 2008.

[116] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[117] S. Theodoridis and K. Koutroumbas, "Pattern recognition (fourth edition)," 2009.

[118] C. Chang and C. Lin, "Libsvm: a library for support vector machines," 2003.

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

[120] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *British Machine Vision Conference (BMVC)*, pp. 384–393, 2002.

[121] Y. Pan, X. Hou, and C. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 800–813, 2011.

[122] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1, pp. 43–72, 2005.

[123] Q. Ye, Q. Huang, W. Gao, and D. Zhao, "Fast and robust text detection in images and video frames," *Image and Vision Computing*, vol. 23, no. 6, pp. 565–576, 2005.

[124] A. Vedaldi and B. Fulkerson, "Vlfeat: an open and portable library of computer vision algorithms. `http://www.vlfeat.org`," 2008.

[125] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–165, 2004.

[126] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[127] K. Fukunaga and L. D.Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

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

[129] W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 37, no. 5, pp. 1382–1389, 2007.

[130] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[131] P. Correia and F. Pereira, "Objective evaluation of video segmentation quality," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 186–200, 2003.

[132] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.

[133] W. Niblack, "An introduction to digital image processing," *New York: Prentice Hall*, 1986.

[134] J. Kittler, J. Illingworth, and J. Foglein, "Threshold selection based on a simple image statistic," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 2, pp. 125–147, 1985.

[135] C. Thillou and B. Gosselin, "Color binarization for complex camera-based images," *Electronic Imaging Conference of the International Society for Optical Imaging*, 2005.

[136] R. Bellman *Dynamic Programming. Courier Dover Publications. ISBN 978-0-486-42809-3.*, 2003.

[137] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[138] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[139] "Random forest `http://en.wikipedia.org/wiki/Random_forest`,"

[140] "Parallel computing `http://en.wikipedia.org/wiki/Parallel_computing`,"

[141] E. Olmedo, J. Calleja, A. Benitez, and M. A. Medina, "Point to point processing of digital images using parallel computing," *International Journal of Computer Science Issues*, vol. 9, no. 3, 2012.

[142] T. Braunl, "Tutorial in data parallel image processing," *Australian Journal of Intelligent Information Processing*, vol. 6, no. 3, pp. 164–174, 2001.

[143] Z. Duan, T. Lei, and H. Fan, "Parallel computing system for image intelligent processing," *Information Technology Journal*, vol. 11, no. 3, pp. 329–333, 2012.

[144] T. Phan, P. Shivakumara, B. Su, and C. Tan, "A gradient vector flow-based method for video character segmentation," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1024–1028, 2011.

[145] "Optical character recognition `http://en.wikipedia.org/wiki/Optical_character_recognition`,"